COMPUTER AND PHYSICAL EXPERIMENTS: DESIGN, MODELING, AND MULTIVARIATE INTERPOLATION

A Thesis
Presented to
The Academic Faculty

by

Lulu Kang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
August, 2010

COMPUTER AND PHYSICAL EXPERIMENTS: DESIGN, MODELING, AND MULTIVARIATE INTERPOLATION

Approved by:

Dr. Roshan Joseph Vengazhiyil, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Jianjun Shi
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. C. F. Jeff Wu
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Nicoleta Serban
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. William Brenneman
Statistics Department
The Procter & Gamble Company

Date Approved: 22 June 2010

To my parents and my family.

ACKNOWLEDGEMENTS

I would like to thank all the people who have taught me, helped me, and inspired me during my doctoral study.

Especially, I would like to express my deepest gratitude to my advisor, Professor Roshan J. Vengazhiyil. He has guided me through my five-year doctoral study with incredible patience, and provided me with tremendous support and encouragement. I am not only inspired by his passion and talents on academic research, but also influenced by his professional ethics and life attitudes, from which I will benefit for my lifetime.

I am extremely grateful to Professor C. F. Jeff Wu. I have worked in his lab for all the years of my doctoral study. He has supported me and taken care of me with great mentorship on both academics and personal life. I have benefitted so much from his broad knowledge and deep insights that I consider myself very fortunate.

I would like to thank Dr. William Brenneman for his valuable mentorship during my two terms of internship at the Procter and Gamble Company, and later his great support for my research and career. I want to acknowledge my sincere gratitude to Professor Jianjun Shi, who not only has served on my dissertation committee, but also gave me so much help and guidance on my career. My thanks also go to Professor Nicoleta Serban for serving on my dissertation committee and providing so many insightful suggestions.

I also want to thank my lab members Dr. Zhiguang Qian, Dr. Tirthankar Dasgupta, Dr. Ying Hung, Dr. Xinwei Deng, Huizhi Xie, Nagesh Adiga, Ba Shan, Chia-Jung Chang, Matthias Tan, and Yijie Wang. It is such an honor to work with these talented people and gain their friendship.

At last but not least, I would like to thank my parents and my family. I cannot ask more from my parents, who have given me unconditional and unlimited love and support despite

of the ups and downs of my life. I also own tremendously to my grandparents, who looked after me before I went college, and loved me with their full heart. I regret so much for not being able to accompany my grandfather during the last period of his life. I miss him. I dedicate my dissertation to my whole family, hoping they would feel happy and proud.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis consists of two parts. The first part focuses on the design and analysis of physical experiments, and the second part on the analysis of computer experiments.

The first part of this thesis contains three works on the design and analysis of physical experiments. In Chapter 1, a new Bayesian optimal design criterion is proposed for robust parameter design experiments, and an algorithm for searching the optimal design is developed. Chapter 2 focuses on the blocked experimental design. A Bayesian approach is developed to overcome the ambiguities existing in the current block design methods. Chapter 3 proposes a new modeling and design strategy for a type of mixture experiments, called *mixture-of-mixtures* experiment. The second part of this thesis introduces two new methodologies for computer experiments. Chapter 4 proposes a new interpolation method called *regression-based inverse distance weighting* method as well as a new method for constructing confidence intervals for the predictions. Chapter 5 first introduces a general nonparametric regression method, called kernel sum regression, and then we show that a particular form of this regression method becomes an interpolator, which can be used to analyze the computer experiments with deterministic outputs.

Chapter 1 deals with the robust parameter design experiments. It is critical to estimate control-by-noise interactions in robust parameter design. This can be achieved by using a cross array, which is a cross product of a design for control factors and another design for noise factors. However, the total run size of such arrays can be prohibitively large. To reduce the run size, single arrays are proposed in the literature, where a modified effect hierarchy principle is used for the optimal selection of the arrays. In Chapter 1, we argue that effect hierarchy principle should not be altered for achieving the robustness objective

of the experiment. We propose a Bayesian approach to develop single arrays which incorporate the importance of control-by-noise interactions without altering the effect hierarchy. The approach is very general and places no restrictions on the number of runs or levels or type of factors or type of designs. A modified exchange algorithm is proposed for finding the optimal single arrays. We also explain how to design experiments with internal noise factors; a topic that has received scant attention in the literature. The advantages of the proposed approach are illustrated using several examples. A paper based on this work is published in *Technometrics* 2009, page 250-261.

The presence of block effects makes the optimal selection of fractional factorial designs a difficult task. The existing frequentist methods try to combine treatment and block wordlength patterns and apply minimum aberration criterion to find the optimal design. However, ambiguities exist in combining the two wordlength patterns and therefore, the optimality of such designs can be challenged. In Chapter 2 we propose a Bayesian approach to overcome this problem. The main technique is to postulate a model and prior distribution to satisfy the common assumptions in blocking and then, to develop an optimal design criterion for the efficient estimation of treatment effects. We apply our method to develop regular, nonregular, and mixed-level blocked designs. Several examples are presented to illustrate the advantages of the proposed method. This work is published on *Journal of Statistical Planning and Inference*, 3319-3328.

Chapter 3 is on mixture-of-mixtures experiments. In this kind of mixture experiments, major components are defined as the components which themselves are mixtures of some other components, called minor components. Sometimes, components are divided into different categories, where each category is called a major component, and the components within a major component become minor components. The special structure of the mixture-of-mixtures experiment makes the design and modeling approaches different from a typical mixture experiment. In Chapter 3, we propose a new model called the major-minor model to overcome some of the limitations of the commonly used multiple-Scheffé model. We

also provide a strategy for designing experiments that are much smaller in size than those based on the the existing methods. We then apply the proposed design and modeling approach to a mixture-of-mixtures experiment conducted to formulate a new potato crisp. This work is tentatively accepted by *Technometrics*.

In Chapter 4 and 5 we shift our interests from physical experiments to computer experiments, which has become increasingly popular in many science and engineering fields. Computer experiments are used to simulate very complex systems; thus there are many challenging issues in analyzing such experiments. In this thesis, we focus on the deterministic computer simulations, where there are no random errors involved in the outputs. Multivariate interpolation methods are used for analyzing such simulation data. In Chapter 4, we proposes a new interpolation method named regression-based inverse distance weighting. It is based on inverse distance weighting (IDW), which is a simple multivariate interpolation method but has poor prediction accuracy. In this chapter we show that the prediction accuracy of IDW can be substantially improved by integrating it with a linear regression model. This new predictor is quite flexible, computationally efficient, and works well in problems having high dimensions and/or large data sets. We also develop a heuristic method for constructing confidence intervals for prediction. This work is tentatively accepted by *Technometrics*

Chapter 5 proposes two analysis methods. The first one is kernel sum regression, which uses an iterative implementation of the simple classic kernel regression. An algorithm is constructed to choose the optimal number of regressions N and the bandwidth parameters based on the generalized cross-validation. The performance of the kernel sum regression is shown to be superior than the simple kernel regression through two examples, thus the extra regressions do improve the prediction. In the second part, we show that as the number of iterations increases to infinity, the kernel sum regression converges to an interpolator, which we name as kernel sum interpolation. It has many interesting connections with the other interpolation methods, such as radial basis function, kriging, as well as the regression-based

inverse distance weighting method introduced in Chapter 4. Compared with these interpolators, kernel sum interpolation is shown to be more robust to the bandwidth parameter.

CHAPTER I

BAYESIAN OPTIMAL SINGLE ARRAYS FOR ROBUST PARAMETER DESIGN

1.1 Introduction

Robust parameter design is a useful technique for quality improvement. The main idea of this technique is to divide the factors in the system into two groups: control factors and noise factors. Then, the settings of the control factors are chosen so that the sensitivity of the response to the noise factors is minimized. This in turn minimizes the transmitted variance to the response from the noise factors and thus, improves the quality. Clearly, robust parameter design can be successful only if the control factors are interacting with the noise factors. Therefore, the experimental design for robustness studies should be capable of estimating the control-by-noise interactions. This aspect makes such designs different from the traditional designs.

Taguchi (1987) proposed to use *cross array*, which is a cross product of a design for control factors (control array) and another design for noise factors (noise array). Thus, in cross arrays, each run in the noise array is repeated for each run in the control array. The advantage of cross arrays is that the interactions between control factors and noise factors can be estimated (see Wu and Hamada 2000, Section 10.7). The disadvantage is that because of the crossing of two arrays, the total run size of the design is very large.

To overcome the drawback of large run size of cross arrays, Welch et al. (1990) and Shoemaker, Tsui, and Wu (1991) proposed to use single arrays. *Single arrays* are fractional factorial designs that incorporate both control factors and noise factors. Single array is a more general concept in the sense that any cross array can be converted to a single array, but in general, the converse is not true. Designing single arrays is more difficult than a

traditional fractional factorial design because of the special importance given to control-by-noise interactions.

The traditional fractional factorial designs are based on the fundamental principle of effect hierarchy. The *effect hierarchy principle* states that lower order effects are more likely to be important than higher order effects and effects of the same order are equally likely to be important (e.g., Wu and Hamada 2000, Section 3.5). To explain this principle in the context of robust parameter design, let us denote a control factor by C and a noise factor by n. Then, according to the hierarchy principle

$$\{C, n\} \succ \{CC, Cn, nn\} \succ \{CCC, CCn, Cnn, nnn\}, \ldots,$$

where $\succ$ denotes "more important than". This does not seem to agree with the objective of robust parameter design, because control-by-noise interactions are considered to be more important than other types of interactions. A quick remedy for this problem seems to be to modify the hierarchy principle so as to satisfy the objectives of robust parameter design. Wu and Zhu (2003) proposed the following modified ordering:

$$\{C, n, Cn\} \succ \{CC, CCn\} \succ \{CCnn, Cnn, nn\}, \ldots,$$

whereas Bingham and Sitter (2003) proposed the ordering

$$\{C, n\} \succ \{Cn\} \succ \{CC, nn\} \succ \{CCn, Cnn\}, \ldots.$$

See also Zhu, Zeng, and Jennings (2007). Although this approach produces reasonably good single arrays, there is a limitation.

Effect order is a property of an engineering system, which is usually unknown to the experimenter. However, it is reasonable to assume that the effect order follows the effect hierarchy principle. This principle should not be changed depending on the objective of an experiment. For example, consider the leaf spring experiment given in Wu and Hamada (2000). Four control factors (high heat temperature, heating time, transfer time, and hold down time) and one noise factor (quench oil temperature) are studied in this experiment.

The interactions among these factors are completely dependent on the underlying physics of the heat treatment process. In general, these interactions tend to follow the effect hierarchy principle. Although the objective is to achieve robustness, we cannot expect that control-by-noise interactions are going to be "more significant" than the other interactions. Of course, they are "more important" in terms of achieving the objectives of the experiment, but there is no way we can alter the already fixed effect order of the heat treatment process. What we can do is to design an experiment that ensures efficient estimation of the "important" effects. In this article, we propose a methodology that can incorporate the importance of control-by-noise interactions without altering the effect hierarchy principle. Note that different from the existing work we distinguish between effect "significance" and "importance"; in fact the definition of effect hierarchy principle previously stated (taken from Wu and Hamada 2000) should have been stated in terms of "significant" effects and not "important" effects.

A nice feature of our approach is that there is no restriction on the number of levels of the factors or the number of runs in the experiment. On the other hand, Bingham and Sitter (2003) and Wu and Zhu (2003) focus only on 2^{p-k} fractional factorial experiments. The response surface designs in Borror, Montgomery, and Myers (2002), Ginsburg and Ben-Gal (2006), and Del Castillo et al. (2007) are general and can entertain different number of factor levels and runs. However, such designs are large and do not incorporate effect hierarchy principle.

The chapter is organized as follows. In Section 1.2, we propose a Bayesian approach to design single arrays that incorporates the robustness objectives while honoring the effect hierarchy principle. Application of the methodology to two-level experiments is discussed in Section 1.3. In this section, we also develop an exchange algorithm that can be used for finding the optimal design. Some examples of optimal single arrays are then provided to compare with the existing designs. In Section 1.4, we generalize our method to mixed-level experiments. In Section 1.5, we discuss how to design experiments with internal noise

factors; a topic that has received scant attention in the literature. We finally conclude this chapter with some remarks in Section 1.6. All proofs are provided in the 1.7.

1.2 Optimal Design Criterion

Our objective is to develop a single array that recognizes the importance of various effects while maintaining effect hierarchy. A Bayesian formulation is suitable for this purpose. We put a prior on the effects so as to reflect effect hierarchy and develop an optimal design criterion that gives more importance to the effects of interest.

Let there be k_C control factors $\boldsymbol{x} = (x_1, x_2, \ldots, x_{k_C})$ and k_n noise factors $\boldsymbol{z} = (z_1, z_2, \ldots, z_{k_n})$. The response y is related to the control and noise factors through

$$y = f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}) + \epsilon, \tag{1.2.1}$$

where $\epsilon \sim N(0, \sigma^2)$ is the random error caused by the unobserved noise factors and $\boldsymbol{\beta}$ is a set of unknown parameters in the model. Although the noise factors are fixed during experimentation, in the actual process they are random. Let $E(z_i) = 0$ and $var(z_i) = \sigma_z^2$ for $i = 1, \ldots, k_n$. By choosing the same variance for all the noise factors, we implicitly assume that the noise factor levels for the experiment are chosen corresponding to the same quantiles of their respective distributions. We also assume them to be independent. When the specific form of the transfer function f is not known, it is convenient to use a linear model: $f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}) = \sum_i \beta_i u_i(\boldsymbol{x}, \boldsymbol{z})$, where u_i's are known functions and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots)$ are unknown parameters. Let the prior distribution for $\boldsymbol{\beta}$ be $N(\boldsymbol{\mu}, \tau^2 \boldsymbol{R})$. We now choose $\boldsymbol{\mu}$ and $\boldsymbol{R}$ in such a way that the effect hierarchy is maintained. This can be easily done using the results in Joseph (2006) and Joseph and Delaney (2007).

Now that effect hierarchy is already incorporated into the model, we only need to focus on deriving an optimal design criterion that satisfy the objectives of robust parameter design. First, approximate the transfer function using a first order Taylor series expansion:

$$f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}) \approx f(\boldsymbol{x}, \boldsymbol{0}, \boldsymbol{\beta}) + \nabla_z f(\boldsymbol{x}, \boldsymbol{0}, \boldsymbol{\beta})' \boldsymbol{z}. \tag{1.2.2}$$

The approximation is reasonable if either the transfer function is approximately linear in $\boldsymbol{z}$ or σ_z is small. Then, the mean and variance of the response are given by

$$E(y) \approx f(\boldsymbol{x}, \boldsymbol{0}, \boldsymbol{\beta}), \tag{1.2.3}$$

$$var(y) \approx \sigma_z^2 \, \nabla_z \, f(\boldsymbol{x}, \boldsymbol{0}, \boldsymbol{\beta})' \, \nabla_z \, f(\boldsymbol{x}, \boldsymbol{0}, \boldsymbol{\beta}) + \sigma^2. \tag{1.2.4}$$

For a nominal-the-best characteristic, the objective is to achieve the mean at target with minimum variation. Adjusting the mean to target is easier than reducing the variation and therefore, estimation of mean is not as important as estimation of variance for such characteristics. In fact, in most cases adjustment factors can be found from engineering knowledge and can be used for adjusting the mean to any desired level (see Joseph 2007). Therefore, here we propose an optimal design criterion to efficiently estimate the variance. However, note that if an adjustment factor is not available or the characteristic is of smaller-the-better or larger-the-better type, then the estimation of mean also becomes important. The optimal design criterion can be easily modified to incorporate this requirement, which is explained in the next section.

As can be seen in (1.2.4), the variance can be efficiently estimated if we can get an efficient estimate of the gradient $\nabla_z f(\boldsymbol{x}, \boldsymbol{0}, \boldsymbol{\beta})$. Let $\widehat{\boldsymbol{\beta}}$ be the Bayes estimate (posterior mean) of $\boldsymbol{\beta}$. Then, the difference between the true gradient and the estimated gradient is

$$d(\boldsymbol{x}, \boldsymbol{\beta}, \widehat{\boldsymbol{\beta}}) = \nabla_z f(\mathbf{x}, \boldsymbol{0}, \boldsymbol{\beta}) - \nabla_z f(\mathbf{x}, \boldsymbol{0}, \widehat{\boldsymbol{\beta}}), \tag{1.2.5}$$

We want this difference to be as small as possible. Therefore, we should minimize

$$\begin{aligned} l(\boldsymbol{x}|\boldsymbol{D}) &= E\{d(\boldsymbol{x}, \boldsymbol{\beta}, \widehat{\boldsymbol{\beta}})' d(\boldsymbol{x}, \boldsymbol{\beta}, \widehat{\boldsymbol{\beta}})\} \\ &= \int_{\boldsymbol{y}} \int_{\boldsymbol{\beta}} d(\boldsymbol{x}, \boldsymbol{\beta}, \widehat{\boldsymbol{\beta}})' d(\boldsymbol{x}, \boldsymbol{\beta}, \widehat{\boldsymbol{\beta}}) p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{D}) d\boldsymbol{\beta} p(\boldsymbol{y}|\boldsymbol{D}) d\mathbf{y}. \end{aligned} \tag{1.2.6}$$

The data $\boldsymbol{y}$ depends on the experimental design $\boldsymbol{D}$. Therefore, we can choose a $\boldsymbol{D}$ such that $l(\boldsymbol{x}|\boldsymbol{D})$ is a minimum. However, $l(\boldsymbol{x}|\boldsymbol{D})$ varies with $\boldsymbol{x}$. We want $l(\boldsymbol{x}|\boldsymbol{D})$ to be small over the entire experimental region for control factors $\mathscr{X}$, which we consider as a discrete space

containing all the candidates points for the experimental design. Therefore, we may choose $\boldsymbol{D}$ that minimizes the average loss over $\mathscr{X}$. Thus, our optimal design criterion is to find $\boldsymbol{D}$ to minimize

$$L(\boldsymbol{D}) = \frac{1}{|\mathscr{X}|} \sum_{\boldsymbol{x} \in \mathscr{X}} l(\boldsymbol{x}|\boldsymbol{D}), \tag{1.2.7}$$

where $|\mathscr{X}|$ is the number of points in the set $\mathscr{X}$. In the following sections, we apply this general method to two- and mixed-level experiments.

1.3 Two-Level Experiments

1.3.1 Optimal design criterion

Consider the case of two-level factors. Let the two levels be -1 and 1. The full linear model can be written as

$$\begin{aligned} f(\boldsymbol{x}, z, \boldsymbol{\beta}) = \mu_0 + \left\{ \beta_0^0 + \sum_{i=1}^{k_C} \beta_i^0 x_i + \ldots + \beta_{12\ldots k_C}^0 x_1 x_2 \ldots x_{k_C} \right\} + \\ + \sum_{j=1}^{k_n} \left\{ \beta_0^j + \sum_{i=1}^{k_C} \beta_i^j x_i + \ldots \right\} z_j + \ldots + \left\{ \beta_0^{2^k_n} + \sum_{i=1}^{k_C} \beta_i^{2^k_n} x_i + \ldots \right\} z_1 z_2 \ldots z_{k_n}. \end{aligned} \tag{1.3.1}$$

Here μ_0 is a constant which is introduced only for simplifying the prior distribution. Let m.e. denote main effects (β_i^0 for $i = 1, \ldots, k_C$ and β_0^j for $j = 1, \ldots, k_n$), 2fi denote two-factor interactions, etc. We use the following prior for $\boldsymbol{\beta}$ proposed in Joseph (2006):

$$\begin{aligned} \beta_0^0 &\sim N(0, \tau^2), \\ \beta_{m.e.} &\sim N(0, \tau^2 r), \\ \beta_{2fi} &\sim N(0, \tau^2 r^2), \\ &\vdots \end{aligned} \tag{1.3.2}$$

and they are all independent, where r is a value between 0 and 1. Thus, $\boldsymbol{\beta} \sim N(\boldsymbol{0}, \tau^2 \boldsymbol{R})$, where $\boldsymbol{R}$ is a diagonal matrix with entries $1, r, \ldots, r, r^2, \ldots, r^{k_C + k_n}$. In this prior, the variances of effects decrease geometrically as the order of the effects increase (note that the means are 0). Therefore, the probability that an effect becomes significant decreases as the order increases, justifying effect hierarchy.

Consider an experiment with m runs. Let $\boldsymbol{D}$ be the $m\times(k_C+k_n)$ design matrix and $\boldsymbol{U_D}$ be the $m\times 2^{k_C+k_n}$ model matrix corresponding to the linear model in (1.3.1). Let $\boldsymbol{y} = (y_1, \ldots, y_m)$ be the data from the experiment. Then, the posterior variance of $\boldsymbol{\beta}$ is given by

$$var(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{D}) = \tau^2\boldsymbol{R} - \tau^2\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}'\left(\boldsymbol{U_D}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}' + \frac{\sigma^2}{\tau^2}\boldsymbol{I}_n\right)^{-1}\boldsymbol{U_D}\boldsymbol{R}. \tag{1.3.3}$$

The objective function in (1.2.7) can be easily computed using the following result.

Proposition 1.3.1. *The objective function defined in* (1.2.7) *is*

$$L(\boldsymbol{D}) = tr(\boldsymbol{A}var(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{D})), \tag{1.3.4}$$

where $\boldsymbol{A}$ is a diagonal matrix, whose diagonal entries are 1 if they correspond to effects involving one and only one noise factor, i.e., $n, Cn, CCn, CCCn, \ldots$, and 0 otherwise.

To understand the foregoing objective function better, consider a small example with two control factors and two noise factors. The model is

$$\begin{aligned} y = \mu_0 + \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 z_1 + \beta_4 z_2 + \beta_5 x_1 z_1 + \beta_6 x_1 z_2 + \beta_7 x_2 z_1 + \beta_8 x_2 z_2 \\ + \beta_9 x_1 x_2 + \beta_{10} z_1 z_2 + \beta_{11} x_1 x_2 z_1 + \beta_{12} x_1 x_2 z_2 + \beta_{13} x_1 z_1 z_2 + \beta_{14} x_2 z_1 z_2 + \beta_{15} x_1 x_2 z_1 z_2 + \epsilon, \end{aligned} \tag{1.3.5}$$

Denote $V_i = var(\beta_i|\boldsymbol{y}, \boldsymbol{D})$. Using (1.3.4), we obtain

$$L(\boldsymbol{D}) = V_3 + V_4 + V_5 + V_6 + V_7 + V_8 + V_{11} + V_{12}.$$

We can see that the objective function includes only the noise main effects (β_3 and β_4) and interactions between control and noise factors ($\beta_5, \beta_6, \beta_7, \beta_8, \beta_{11}$, and β_{12}). The control main effects (β_1 and β_2) and control-by-control interaction (β_9) affect only the mean and therefore, are not needed for minimizing variance. The higher order noise effects (β_{13}, β_{14}, and β_{15}) are neglected in computing the variance. Thus, the optimal design criterion includes only those effects that are important for achieving robustness. Note that effect hierarchy is already incorporated in the prior information and therefore, the foregoing Bayesian method

is able to separately incorporate both "importance" and "significance" of effects, which is not the case with the existing frequentist methods.

Because the loss function $L(\boldsymbol{D})$ does not have any interpretation, it is more convenient to transform it to the following utility function

$$U(\boldsymbol{D}) = \frac{tr\left(\boldsymbol{A}var(\boldsymbol{\beta})\right) - tr\left(\boldsymbol{A}var(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{D})\right)}{tr\left(\boldsymbol{A}var(\boldsymbol{\beta})\right)}. \tag{1.3.6}$$

Thus, if the design is completely noninformative ($m = 0$), then $U(\boldsymbol{D}) = 0$ and if it is completely informative ($m = 2^{k_C+k_n}$), then $U(\boldsymbol{D}) = 1$. Therefore, the value of $U(\boldsymbol{D})$ clearly shows how good the design is. We obtain

$$U(\boldsymbol{D}) = \frac{tr\left(\boldsymbol{A}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}'\left(\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}' + \sigma^2/\tau^2\boldsymbol{I}_n\right)^{-1}\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\right)}{tr(\boldsymbol{A}\boldsymbol{R})}. \tag{1.3.7}$$

The optimal design can be found by maximizing $U(\boldsymbol{D})$.

Note that as mentioned in the previous section, there are cases where the estimation of the mean is also important. In such cases, we can define a utility function for the mean as in (1.3.6), where the parameters $\boldsymbol{\beta}$ correspond to $C, CC, CCC, \ldots$. Denote the utility functions for the mean and variance by $U_{mean}(\boldsymbol{D})$ and $U_{var}(\boldsymbol{D})$, respectively. Then, we can find the optimal design by maximizing $U(\boldsymbol{D}) = \lambda U_{mean}(\boldsymbol{D}) + (1-\lambda)U_{var}(\boldsymbol{D})$, where $\lambda \in [0, 1]$ is chosen depending on the relative importance of mean and variance for that particular problem. In this work, we focus on the case of $\lambda = 0$.

There are several parameters (σ^2, τ^2, and r) that need to be specified before we can find the optimal design. The problem is, in most cases, we do not know the "best" values of these parameters before conducting the experiment. Therefore, some reasonable and meaningful values should be chosen for designing the experiment. As argued by Joseph (2006), by assuming a high signal-to-noise ratio we can neglect the ratio σ^2/τ^2. Then, the criterion reduces to maximizing $tr(\boldsymbol{A}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}'(\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}')^{-1}\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R})$. Note that we omitted the denominator in (1.3.7), because it does not depend on $\boldsymbol{D}$. Now we only need to specify the value of r. Li, Sudarsanam, and Frey (2006) did a meta-analysis of 113 data sets from

published experiments. They found that the median strength of two-factor interactions is 1/4th of the median strength of main effects and the median strength of three-factor interactions is half of the median strength of two-factor interactions. Based on their finding, we think it is reasonable to choose $r = 1/3$ in the absence of any prior knowledge about the particular product/process under investigation.

Having specified all the unknown parameters in the objective function, we can proceed to find the optimal design for a given number of runs m. The Bayesian approach is capable of producing an optimal design for any m. This is definitely an advantage over the frequentist methods. However, we should realize that a Bayesian approach cannot do any magic. If the run size is too small, then the results heavily rely on the prior information as there is little information from the experiments. Therefore, even though we use a Bayesian approach, it is important to choose a reasonable number of runs for the experiment. We recommend that the design should be capable of estimating at least the grand mean, control main effects, noise main effects, and two-factor control-by-noise interactions in the frequentist sense. A necessary condition for this is

$$m \geq 1 + k_C + k_n + k_C k_n = (1 + k_n)(1 + k_C). \tag{1.3.8}$$

Note that although our objective function includes only the n and Cn effects, we additionally included I and C effects when calculating the minimum number of runs. This is because, if the grand mean or the control main effects are not estimable, i.e., $I = C$ or $C_1 = C_2$, then $n = Cn$ or $C_1 n = C_2 n$, which should not happen.

1.3.2 Exchange algorithm

Exchange algorithms are the most common form of computer design search algorithms. See Nguyen and Miller (1992) for a review of some early versions of exchange algorithm for constructing D-optimal designs. The basic idea of exchange algorithm is to randomly choose an initial design and perform row-wise exchanges of some points from a candidate set of feasible design points until the objective function is optimized. There have been many

new types of exchange algorithms developed for some other optimal design criteria, such as columnwise-pairwise exchange algorithm (Li and Wu, 1997) and coordinate-exchange algorithm (Meyer and Nachtsheim, 1995).

Although these algorithms are very effective in optimizing most frequentist based optimal design criteria, it is hard to implement them with our Bayesian optimal design criterion. Therefore, we propose a modified exchange algorithm that takes advantage of the special matrix form in our objective function. Moreover, instead of randomly choosing an initial design, we construct our initial design in a "greedy" fashion to accelerate the convergence of the later "exchange" part of the algorithm. In the simulations we carried out, this modified algorithm is found to perform quite well.

For this section, $\boldsymbol{x}$ is used for generically denoting a point in the design space containing *both* control and noise factors. Let $\boldsymbol{D} = \{\mathbf{x}_1, \ldots, \boldsymbol{x}_l\}$ be the set of points that are currently selected as design points, where $l \leq m$. The remaining $2^{k_C+k_n} - l$ points of the design space form the candidate set. The model matrix generated for the points in $\boldsymbol{D}$ is denoted as $\boldsymbol{U_D}$. If a new point $\boldsymbol{x}$ is added into D, then it introduces a new row in the model matrix $\boldsymbol{U_D}$. Denote the new row in $\boldsymbol{U_D}$ by $\boldsymbol{F}' = (1, u_1, u_2, \ldots)'$ which consists of all the main effects and interactions. The new model matrix is $\tilde{\boldsymbol{U}}_\mathbf{D} = (\boldsymbol{U}'_\mathbf{D}, \boldsymbol{F})'$.

Let $\boldsymbol{M} = \boldsymbol{R}\boldsymbol{U}'_\mathbf{D}(\boldsymbol{U_D}\boldsymbol{R}\boldsymbol{U}'_\mathbf{D})^{-1}\boldsymbol{U_D}\boldsymbol{R}$, and $\boldsymbol{M}_x = \boldsymbol{R}\tilde{\boldsymbol{U}}'_\mathbf{D}(\tilde{\boldsymbol{U}}_\mathbf{D}\boldsymbol{R}\tilde{\boldsymbol{U}}'_\mathbf{D})^{-1}\tilde{\boldsymbol{U}}_\mathbf{D}\boldsymbol{R}$. As shown in the Appendix 1.7, we can update $\boldsymbol{M}_x$ in the following way:

$$\boldsymbol{M}_x = \boldsymbol{M} + \frac{1}{d}\left((\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F}\right)\left((\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F}\right)', \tag{1.3.9}$$

where $d = \boldsymbol{F}'(\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F}$. Thus the objective function $tr(\boldsymbol{A}\boldsymbol{M})$ is increased to $tr(\boldsymbol{A}\boldsymbol{M}_x)$ by $\Delta(\boldsymbol{x}, \boldsymbol{D})$, where

$$\begin{aligned}\Delta(\boldsymbol{x}, \boldsymbol{D}) &= \frac{1}{d}((\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F})'\boldsymbol{A}((\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F}) \\ &= \frac{1}{d}\sum_{i:A_{ii}=1}((\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F})_i^2 .\end{aligned} \tag{1.3.10}$$

We do not need to worry about the numerical stability of the algorithm due to two observations: (i) $\boldsymbol{R} - \boldsymbol{M}$ is a positive definite matrix so that $d > 0$ and (ii) as long as no

points in D are replicated, $\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}'$ is invertible. Because of (ii), we only search for designs where none of the rows are replicated. However, replication is useful for estimating the unknown variance σ^2. Therefore, if we can afford replications, then we simply concatenate the optimal unreplicated design as many times as the number of replicates. The algorithm is given below.

step 0. Generate the complete design space which contains $2^{k_C+k_n}$ points, the correlation matrix $\boldsymbol{R}$, and specify the matrix $\boldsymbol{A}$.

step 1. Randomly select m_0 points into design point set $\boldsymbol{D}$. Update $\boldsymbol{U}_{\mathbf{D}}$, $\boldsymbol{M}$, and objective value is $tr(\boldsymbol{A}\boldsymbol{M})$.

step 2. Initial design: evaluate the increment $\Delta(\boldsymbol{x}, \boldsymbol{D})$ in (1.3.10) for every point in the current candidate set. Add the candidate point that gives the largest $\Delta(\boldsymbol{x}, \boldsymbol{D})$ into $\boldsymbol{D}$. Then update $\boldsymbol{D}$ and the candidate set. There is no need to update $\boldsymbol{U}_{\mathbf{D}}$. Instead, we can directly update $\boldsymbol{M}$ to $\boldsymbol{M}_x$ by using (1.3.9). Repeat this step until m points have been selected into $\boldsymbol{D}$.

step 3. Exchange part: for each point $\boldsymbol{x}_i$ in $\boldsymbol{D}$, denote $\boldsymbol{D}_{-i} = \boldsymbol{D}\backslash\boldsymbol{x}_i$. Compute

$$\boldsymbol{M}_{-i} = \boldsymbol{R}\boldsymbol{U}_{\mathbf{D}_{-i}}'(\boldsymbol{U}_{\mathbf{D}_{-i}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}_{-i}}')^{-1}\boldsymbol{U}_{\mathbf{D}_{-i}}\boldsymbol{R}.$$

Evaluate the $\Delta(\boldsymbol{x}, \boldsymbol{D}_{-i})$ for every candidate point $\boldsymbol{x}$. Choose the $\boldsymbol{x}^*$ that has $\Delta(\boldsymbol{x}^*, \boldsymbol{D}_{-i}) > \Delta(\boldsymbol{x}_i, \boldsymbol{D}_{-i})$ and also $\Delta(\boldsymbol{x}^*, \boldsymbol{D}_{-i}) \geq \Delta(\boldsymbol{x}, \boldsymbol{D}_{-i})$ for any $\boldsymbol{x}$ in the candidate set. Exchange $\boldsymbol{x}^*$ with $\boldsymbol{x}_i$. Then update $\boldsymbol{D}$ and the candidate set. Update the objective function value to $tr(\boldsymbol{A}\boldsymbol{M}_{-i}) + \Delta(\boldsymbol{x}^*, \boldsymbol{D}_{-i})$.

step 4. Repeat *step 3* until the objective value has been stabilized.

Like all exchange algorithms, the optimal design returned can be a local optimum. So we begin with a randomly selected m_0 points. If m_0 is too small, little randomness is introduced and it is hard to escape from a local optimum; if m_0 is close to m, then Step 2

plays such a small role that the exchange part converges very slowly. Our simulation study shows that $m_0 \in \{m/4, m/3, m/2\}$ are good choices. Many trials of the algorithm should be employed and the best design should be returned. When the number of factors is large, the storage of the model matrix $\boldsymbol{U}$ can be an issue. In such cases, the size of $\boldsymbol{U}$ can be reduced by using only up to two- or three-factor interactions. The code for generating the optimal design is available from the authors upon request.

1.3.3 Examples

Example 1: Miller et al. (1993) studied an experiment on the geometric distortion of drive gears in a heat treatment process. There are five control factors: carbon potential (A), operating model (B), last zone temperature (C), quench oil temperature (D), and quench oil agitation (E) and three noise factors: furnace track (a), tooth size (b), and part positioning (c) (denoted as F, G, and H in the original paper). The original experiment used a cross array design with $16 \times 8 = 128$ runs. Miller et al. (1993) reanalyzed the data using single arrays of sizes 64 and 32 and obtained essentially the same results as the original experiment. Here we consider a much smaller single array. From (1.3.8), $m \geq 24$. Therefore, we choose $m = 24$. With $r = 1/3$, we obtain the optimal design that maximizes $U(\boldsymbol{D})$. The optimal design ($\boldsymbol{D}_1$) is shown in Table 1.3.1.

Consider an alternative design $\boldsymbol{D}_2$, which is a D-optimal design for the model containing only control and noise main effects and two-factor control-by-noise interactions (we created this using the software JMP 7.0). Figure 1.3.1 shows the relative efficiency $U(\boldsymbol{D}_2)/U(\boldsymbol{D}_1)$ for different values of r. We can see that $\boldsymbol{D}_1$ is uniformly better than $\boldsymbol{D}_2$ for all values of $r \in (0, 1)$. Figure 1.3.2 shows the relative efficiency for different values of $\sigma^2/\tau^2 \in [0, 5]$ with $r = 1/3$. $\boldsymbol{D}_1$ is still uniformly better than $\boldsymbol{D}_2$, but the improvement diminishes as σ^2/τ^2 increases. We also studied the effect of number of runs. The utility function of the optimal design $U(\boldsymbol{D}^*)$ for $m = 1, 2, \ldots, 256$ with $r = 1/3$ and $\sigma^2/\tau^2 = 0$ is shown in Figure 1.3.3. It can be seen that the value of $U(\boldsymbol{D}^*)$ increases with m and reaches

Table 1.3.1: The optimal design $\boldsymbol{D}_1$ and D-optimal design $\boldsymbol{D}_2$ for Example 1

	Optimal Design ($\boldsymbol{D}_1$)								D-Optimal Design ($\boldsymbol{D}_2$)							
run	A	B	C	D	E	a	b	c	A	B	C	D	E	a	b	c
1	−1	1	−1	−1	−1	−1	−1	−1	−1	−1	−1	1	−1	−1	−1	1
2	−1	−1	1	1	−1	−1	−1	−1	−1	−1	1	−1	1	−1	1	−1
3	1	1	1	−1	1	−1	−1	−1	−1	−1	−1	−1	−1	1	1	1
4	1	−1	−1	1	−1	1	−1	−1	−1	−1	−1	1	1	1	1	1
5	−1	−1	1	−1	1	1	−1	−1	−1	1	1	−1	−1	−1	1	−1
6	−1	1	−1	1	1	1	−1	−1	1	1	1	−1	1	1	−1	−1
7	1	−1	−1	1	−1	−1	1	−1	1	−1	−1	1	−1	1	−1	−1
8	−1	−1	1	−1	1	−1	1	−1	−1	−1	−1	−1	1	1	−1	−1
9	−1	1	−1	1	1	−1	1	−1	1	−1	1	−1	1	−1	−1	1
10	−1	1	−1	−1	−1	1	1	−1	−1	−1	1	1	−1	1	−1	−1
11	−1	−1	1	1	−1	1	1	−1	−1	1	1	1	1	−1	1	−1
12	1	1	1	−1	1	1	1	−1	−1	1	−1	−1	−1	1	−1	−1
13	1	−1	−1	1	−1	−1	−1	1	1	1	−1	−1	1	−1	1	−1
14	−1	−1	1	−1	1	−1	−1	1	−1	1	1	1	−1	1	1	1
15	−1	1	−1	1	1	−1	−1	1	−1	−1	1	−1	1	1	1	1
16	−1	1	−1	−1	−1	1	−1	1	1	1	−1	1	1	−1	−1	1
17	−1	−1	1	1	−1	1	−1	1	−1	1	1	1	1	−1	−1	1
18	1	1	1	−1	1	1	−1	1	−1	1	−1	−1	−1	−1	−1	1
19	−1	1	−1	−1	−1	−1	1	1	1	−1	1	1	−1	−1	1	−1
20	−1	−1	1	1	−1	−1	1	1	−1	−1	−1	1	−1	−1	1	−1
21	1	1	1	−1	1	−1	1	1	−1	1	−1	1	1	1	−1	−1
22	1	−1	−1	1	−1	1	1	1	1	−1	1	1	−1	1	1	1
23	−1	−1	1	−1	1	1	1	1	1	1	−1	−1	1	1	1	1
24	−1	1	−1	1	1	1	1	1	1	1	1	1	−1	−1	−1	1

1 when the design is full factorial.

For a more fair comparison of the two designs we use a frequentist criterion. Consider the average absolute correlation of an effect with the other effects (main effects and two-factor interactions). Table 1.3.2 shows the average of the average absolute correlations of effects within each group. Clearly the effects of the type n and Cn are less correlated with the other effects in $\boldsymbol{D}_1$ than $\boldsymbol{D}_2$ and therefore, $\boldsymbol{D}_1$ is a better design for estimating the noise main effects and control-by-noise interactions.

Example 2: Suppose an experiment involves five control factors A, B, C, D, E and one noise factor a. From (1.3.8), $m \geq 12$. This time let us choose $m = 16$. Interestingly, for all $r \in (0, 1)$, the optimal design returned by the algorithm is a fractional factorial design with

Table 1.3.2: Average absolute correlations of effects for $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ in Example 1

Design	C	n	Cn	CC	nn
Optimal design ($\boldsymbol{D}_1$)	0.1674	0.0152	0.0515	0.1722	0.0606
D-Optimal design ($\boldsymbol{D}_2$)	0.1589	0.1496	0.1540	0.2137	0.1496

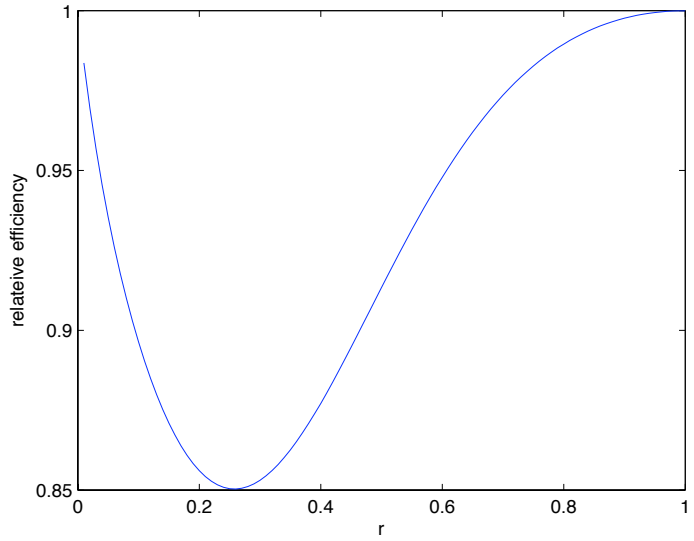


Figure 1.3.1: Relative Efficiency of D_2 to D_1 ($\sigma^2/\tau^2 = 0$).

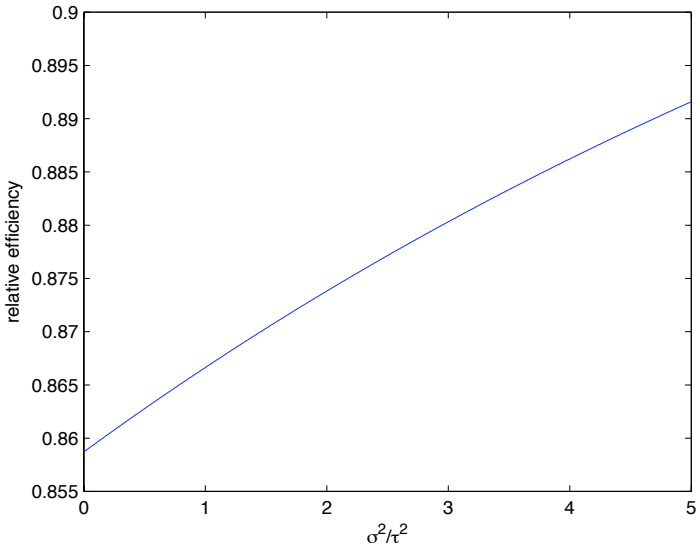


Figure 1.3.2: Relative Efficiency of D_2 to D_1 ($r = 1/3$).

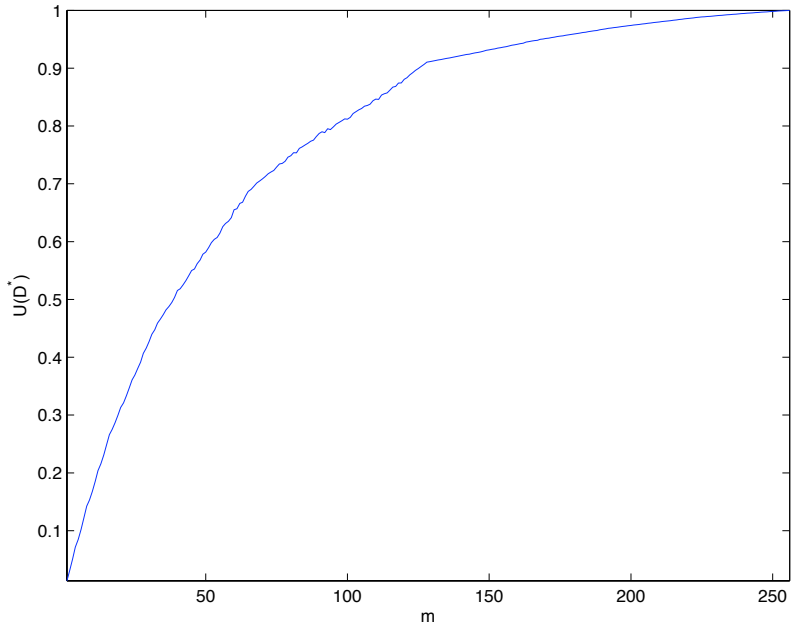


Figure 1.3.3: Utility function value of optimal designs with $m = 1, \ldots, 256$ ($r = 1/3$, $\sigma^2/\tau^2 = 0$).

defining contrast subgroup: $I = ADE = ABC = BCDE$. Furthermore, it can be converted to a cross array because the noise factor does not appear in the defining contrast subgroup. Denote this design by $\boldsymbol{D}_1$.

The Minimum J-aberration single array for this setting given by Wu and Zhu (2003) is $\boldsymbol{D}_2$: $I = ABD = aACE = aBCDE$. It can be shown numerically that $U(\boldsymbol{D}_1) > U(\boldsymbol{D}_2)$ for all $r \in (0, 1)$ and therefore, $\boldsymbol{D}_1$ is uniformly better than $\boldsymbol{D}_2$. We also compared the two designs from a frequentist point of view. Let N_C denote the number of clear control main effects, N_n the number of clear noise main effects, and so on. The estimation capacity of $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ in terms of clear effects is summarized in Table 1.3.3. Although $\boldsymbol{D}_2$ has more number of clear effects, it has three less clear control-by-noise interactions than $\boldsymbol{D}_1$. Therefore, $\boldsymbol{D}_1$ is a much better design for studying robustness.

Table 1.3.3: Comparison of Estimation Capacity of $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ for Example 2.

Design	Clear Effects	N_C	N_n	N_{CC}	N_{Cn}
Optimal design ($\boldsymbol{D}_1$)	a, aA, aB, aC, aD, aE	0	1	0	5
Wu-Zhu ($\boldsymbol{D}_2$)	a, C, E, aB, aD, BC, BE, CD, DE	2	1	4	2

1.4 Mixed-Level Experiments

Experiments with mixed two- and three-level factors are very common in practice. In this section we explain how to design such experiments. The three levels are chosen only for control factors because the transfer function is assumed to be approximately linear with respect to the noise factors.

Suppose there are k_n two-level noise factors, k_{C2} two-level and k_{C3} three-level control factors. Denote the three levels by -1, 0, and 1. As in (1.3.1), the transfer function can be written as

$$f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}) = \mu_0 + \boldsymbol{\beta}'\boldsymbol{u}, \tag{1.4.1}$$

where $\boldsymbol{u} = (1, u_1, u_2, \ldots)'$ are functions of $\boldsymbol{x}$ and $\boldsymbol{z}$ representing their main effects and interactions. There are a total of $p = 2^{k_n+k_{C2}}3^{k_{C3}}$ u-variables. The main task is to postulate a prior distribution for the parameters that satisfy effect hierarchy and then, to derive the optimal design criterion. The first part is made simple by using the functionally induced priors in Joseph and Delaney (2007).

Note that for two-level experiments, we used the prior

$$\boldsymbol{\beta} \sim N(0, \tau^2\boldsymbol{R}), \quad \text{where } \boldsymbol{R} = diag\{1, r, \ldots, r, r^2, \ldots, r^2, \ldots, r^{k_C+k_n}\}.$$

The prior variance can be equivalently written using Kronecker products as (Joseph and Delaney 2007):

$$var(\boldsymbol{\beta}) = \sigma_0^2 \bigotimes_{j=1}^{k_C+k_n} \boldsymbol{U}_j^{-1}\boldsymbol{\Psi}_j(\boldsymbol{U}_j^{-1})'. \tag{1.4.2}$$

where $\sigma_0^2 = \tau^2(1+r)^{k_C+k_n}$, $r = (1-\rho)/(1+\rho)$,

$$\boldsymbol{U}_j = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \text{ and } \boldsymbol{\Psi}_j = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}. \tag{1.4.3}$$

The $\boldsymbol{U}_j$ and $\boldsymbol{\Psi}_j$ are the model matrix and correlation matrix of the jth two-level factor. The form in (2.5.1) can be easily extended to specify the prior for mixed-level factors. Thus $\boldsymbol{\beta} \sim N(\mathbf{0}, \tau^2\boldsymbol{R})$, where

$$\tau^2\boldsymbol{R} = \sigma_0^2 \bigotimes_{j=1}^{k_n} \boldsymbol{U}_j(z_j)^{-1}\boldsymbol{\Psi}_j(z_j)(\boldsymbol{U}_j^{-1}(z_j))' \bigotimes_{j=1}^{k_{C2}+k_{C3}} \boldsymbol{U}_j(x_j)^{-1}\boldsymbol{\Psi}_j(x_j)(\boldsymbol{U}_j(x_j)^{-1})'. \tag{1.4.4}$$

The matrices $\boldsymbol{U}_j$ and $\boldsymbol{\Psi}_j$ for two-level factors are given in (2.5.2). For a three-level factor the choice of these matrices depends on whether the factor is qualitative or quantitative.

Joseph and Delaney (2007) suggested using Helmert coding for qualitative factors and orthogonal polynomial coding for quantitative factors. For the case of three-level factors, these two coding schemes lead to identical model matrix:

$$\boldsymbol{U}_j = \begin{pmatrix} 1 & -\sqrt{\frac{3}{2}} & \sqrt{\frac{1}{2}} \\ 1 & 0 & -\sqrt{2} \\ 1 & \sqrt{\frac{3}{2}} & \sqrt{\frac{1}{2}} \end{pmatrix}. \tag{1.4.5}$$

An isotropic correlation function is recommended for qualitative factors and a Gaussian correlation function for quantitative factors. Thus, the correlation matrices for qualitative and quantitative factors are

$$\boldsymbol{\Psi}_j = \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix} \text{ and } \boldsymbol{\Psi}_j = \begin{pmatrix} 1 & \rho & \rho^4 \\ \rho & 1 & \rho \\ \rho^4 & \rho & 1 \end{pmatrix}, \tag{1.4.6}$$

respectively.

Let $\boldsymbol{D}$ be the $m \times (k_{C2} + k_{C3} + k_n)$ design matrix and $\boldsymbol{U}_{\mathbf{D}}$ the $m \times p$ model matrix. The following result extends the Proposition 1.3.1 to the more general case of mixed-levels.

Proposition 1.4.1. *If we choose the model matrices as in (2.5.2) and (2.5.3), then the loss function in* (1.2.7) *becomes*

$$L(\boldsymbol{D}) = tr\left(\boldsymbol{A}var(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{D})\right), \tag{1.4.7}$$

where $\boldsymbol{A}$ is a diagonal matrix whose diagonal entries corresponding to effects containing one and only one noise factor are 1, and 0 otherwise and $var(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{D})$ is given in (1.3.3).

Define $U(\boldsymbol{D})$ as in (1.3.7). As before, we neglect the term σ^2/τ^2 in $U(\boldsymbol{D})$ by assuming a high signal-to-noise ratio. We also need to choose a value for ρ for computing $U(\boldsymbol{D})$. For a two-level factor, we know that $r = 1/3$ is a reasonable choice. Thus, from the relation $r = (1-\rho)/(1+\rho)$, we obtain $\rho = 1/2$. This value for ρ will be used for both qualitative and quantitative factors. For the number of runs, we use the same argument as in (1.3.8). We obtain $m \geq (1 + k_n)(1 + k_{C2} + 2k_{C3})$.

Example 3: Suppose an experiment contains one two-level noise factor a, two three-level qualitative control factors A and B, and two three-level quantitative control factors C and D. The number of runs $m \geq 18$. Therefore, we choose $m = 18$. The optimal design $\boldsymbol{D}_1$ is given in Table 1.4.

Table 1.4.1: The design $\boldsymbol{D}_1$ and $\boldsymbol{D}_3$ for Example 3.

	$\boldsymbol{D}_1$					$\boldsymbol{D}_3$				
run	A	B	C	D	a	A	B	C	D	a
1	−1	−1	−1	−1	−1	−1	0	−1	1	−1
2	−1	−1	−1	−1	1	0	−1	1	1	1
3	0	1	0	−1	−1	1	0	0	1	1
4	0	1	0	−1	1	1	−1	−1	0	1
5	−1	0	1	−1	−1	1	1	1	1	−1
6	−1	0	1	−1	1	0	0	1	−1	−1
7	1	0	−1	0	−1	0	1	0	0	1
8	1	0	−1	0	1	−1	−1	1	0	−1
9	1	−1	0	0	−1	1	1	1	−1	1
10	1	−1	0	0	1	1	−1	−1	−1	−1
11	0	−1	1	0	−1	0	0	−1	−1	1
12	0	−1	1	0	1	−1	−1	0	−1	1
13	0	1	−1	1	−1	−1	1	0	−1	−1
14	0	1	−1	1	1	−1	1	−1	1	1
15	−1	0	0	1	−1	−1	0	1	0	1
16	−1	0	0	1	1	0	−1	0	1	−1
17	1	1	1	1	−1	0	1	−1	0	−1
18	1	1	1	1	1	1	0	0	0	−1

Note that the optimal design is a balanced design, but it is not orthogonal. For comparison, consider another two designs. Design $\boldsymbol{D}_2$ is constructed from the famous orthogonal array $OA(18, 2^1 3^7)$ (Taguchi 1987), in which factors a and $A \sim D$ are assigned to the

Table 1.4.2: Average absolute correlations of effects for $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$in Example 3

Design	n	$C_l n$	$C_q n$	C_l	C_q	$C_l C_l$	$C_l C_q$	$C_q C_q$
$\boldsymbol{D}_1$	0	0.0531	0.0427	0.4772	0.4132	0.5441	0.5191	0.5388
$\boldsymbol{D}_2$	0	0.2364	0.2253	0.2547	0.2032	0.3623	0.3299	0.3727
$\boldsymbol{D}_3$	0.1972	0.2740	0.2799	0.3260	0.2635	0.3802	0.3691	0.3930

columns $1 \sim 5$. Design $\boldsymbol{D}_3$ is the D-optimal design for the model containing the 18 effects (main effects and two-factor control-by-noise interactions). For $\rho = 1/2$ we obtain, $U(\boldsymbol{D}_2) = 0.2467 < U(\boldsymbol{D}_3) = 0.2569 < U(\boldsymbol{D}_1) = 0.3679$ and therefore, $\boldsymbol{D}_1$ is a much better design than both $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$. In fact, $\boldsymbol{D}_1$ is uniformly better than $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$ for all $\rho \in (0, 1)$. Table 1.4.2 compares the average absolute correlation values for each category of effects. We can see that the effects n, $C_l n$, and $C_q n$ have the smallest average absolute correlation with the other effects in $\boldsymbol{D}_1$ compared to $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$. Thus $\boldsymbol{D}_1$ is a better design than $\boldsymbol{D}_2$ and $\boldsymbol{D}_3$ for estimating the response variance.

For the simplicity of the exposition, we have focused only on the mixed two- and three-level factors. The method can be easily generalized to factors with any number of levels using the functionally induced priors in Joseph and Delaney (2007) and following the derivation of Proposition 1.4.1.

1.5 Factors with Internal Noise

There are factors such as temperature in a heat treatment process and current in a welding process that fluctuate around a nominal value. The nominal values of such factors can be easily controlled and therefore, are considered as control factors, whereas the variations around the nominal values are uncontrollable and are called *internal noise* factors (Taguchi 1987). Internal noise factors arise quite often in experiments, but surprisingly have received little attention in the experimental design literature. We propose two important guiding principles for designing experiments with internal noise factors:

1. Factors with internal noise should be experimented with at least three levels.

2. For factors with internal noise, the interactions among those factors and the interactions with the other control and external noise factors are important.

To explain these principles, consider a factor (T_1) with internal noise. We can represent this factor as $T_1 = t_1 + z_1$, where the nominal value t_1 is a control factor and z_1 is the internal noise factor (see Joseph 2003 and 2008 for examples). The first principle is needed, because it will help to estimate the nonlinearity of the response with respect to T_1. By exploiting this nonlinearity, we can understand the interactions between t_1 and z_1 and thus, achieve robustness (note that $T_1^2 = t_1^2 + z_1^2 + 2t_1z_1$ contains the interaction t_1z_1). Of course, the nonlinearity can be estimated only if the factor is varied in at least three levels. The need of second principle can be explained as follows. Consider another control factor x_1 and an *external noise* factor z_2. By entertaining the interaction T_1x_1, we can study the interaction between x_1 and z_1 and by entertaining the interaction T_1z_2, we can study the interaction between t_1 and z_2. Both these interactions are important for achieving robustness. Similarly, the interaction between two factors with internal noise is also important.

Although a factor with internal noise can be conveniently represented as the sum of a control factor and an internal noise factor, the experimental design and the optimal design criterion do not reduce to the cases that we have discussed so far. This is because the internal noise factors are not systematically varied in a robust parameter design experiment. Thus, a design of experiment with T_1 will only specify the values of t_1 but not z_1. However, the actual values of T_1 during the experiment differ from t_1 due to the internal noise. To distinguish these two cases, denote $\bar{\boldsymbol{D}}$ as the design containing the values of t_1 and $\boldsymbol{D}$ as the design containing the values of T_1. Before the experiment is conducted, $\boldsymbol{D}$ can be viewed as a random matrix such that $E(\boldsymbol{D}) = \bar{\boldsymbol{D}}$.

For example, suppose temperature is a factor that cannot be controlled precisely in a manufacturing process. The existing specification for temperature is say, $200 \pm 10\ ^oF$. Now for experimenting with this factor, we should choose at least three levels for the nominal

value, say 170, 200, and 230 ^{o}F. These are the three settings for t_1. Although the temperature may vary in ± 10 ^{o}F from the nominal values during the usual operation of the process, it might be possible to control the temperature more precisely during experimentation, say in ± 5 ^{o}F. In any case, it is a good idea to measure the temperature during the experiment, which can be used for model fitting (Freeny and Nair 1992). Here the planned design matrix in which the temperature takes the values $170, 200$, and 230 is denoted as $\bar{\boldsymbol{D}}$, whereas the matrix containing the actual values of temperature is one realization of the random matrix $\boldsymbol{D}$. Consider the following example.

Example 4: Let T_1 be a factor with internal noise, x_1 a two-level control factor, and z_2 an external noise factor. The internal noise and external noise are independent and follow $N(0, \sigma_z^2)$. Because T_1 has internal noise, we choose three levels for it. The linear and quadratic effects of T_1 using orthogonal polynomial coding are

$$T_{1l} = \sqrt{\frac{3}{2}} T_1 \quad \text{and} \quad T_{1q} = \frac{3}{\sqrt{2}}(T_1^2 - \frac{2}{3}).$$

Note that this coding gives us the same model matrix in (2.5.3) when T_1 takes the values -1, 0, and 1. The transfer function is

$$\begin{aligned} f(x_1, T_1, z_2, \boldsymbol{\beta}) = \mu_0 + \beta_0 + \beta_1 T_{1l} + \beta_2 x_1 + \beta_3 z_2 + \beta_4 T_{1l} x_1 + \beta_5 T_{1l} z_2 + \beta_6 x_1 z_2 \qquad (1.5.1) \\ + \beta_7 T_{1q} + \beta_8 T_{1q} x_1 + \beta_9 T_{1q} z_2 + \beta_{10} T_{1l} x_1 z_2 + \beta_{11} T_{1q} x_1 z_2. \end{aligned}$$

Substituting $T_1 = t_1 + z_1$ into the model and neglecting the small terms involving σ_z^4, we can approximate the variance by

$$\begin{aligned} Var(f) \approx \left(\beta_1 \sqrt{\frac{3}{2}} + \beta_4 \sqrt{\frac{3}{2}} x_1 + \beta_7 3\sqrt{2} t_1 + \beta_8 3\sqrt{2} t_1 x_1 \right)^2 \sigma_z^2 \qquad (1.5.2) \\ + \left(\beta_3 + \beta_5 \sqrt{\frac{3}{2}} t_1 + \beta_6 x_1 + \beta_9 \frac{3}{\sqrt{2}}(t_1^2 - \frac{2}{3}) + \beta_{10} \sqrt{\frac{3}{2}} t_1 x_1 + \beta_{11} \frac{3}{\sqrt{2}}(t_1^2 - \frac{2}{3}) x_1 \right)^2 \sigma_z^2, \end{aligned}$$

which is easily seen as equal to:

$$\left(\frac{\partial f(x_1, t_1, \mathbf{0}, \boldsymbol{\beta})}{\partial T_1} \right)^2 \sigma_z^2 + \left(\frac{\partial f(x_1, t_1, \mathbf{0}, \boldsymbol{\beta})}{\partial z_2} \right)^2 \sigma_z^2.$$

The loss function can be derived as in Section 1.2 with the only exception that the design $\boldsymbol{D}$ is random and the objective is to find $E(\boldsymbol{D}) = \bar{\boldsymbol{D}}$. Therefore, we should minimize $E\{L(\boldsymbol{D})\}$ with respect to $\bar{\boldsymbol{D}}$. To simplify the optimization we approximate $E\{L(\boldsymbol{D})\}$ by $L(\bar{\boldsymbol{D}})$. This approximation is reasonable because the internal noise factors can be controlled precisely during the experiment.

Let T_j be the jth factor with internal noise. Define

$$i_j = \begin{cases} 1, & \text{if the } i\text{th effect contains } T_{jl} \text{ or } T_{jq}, \\ 0, & \text{otherwise.} \end{cases} \tag{1.5.3}$$

For $i_j = 1$, define

$$\delta_{i,j} = \begin{cases} 1, & \text{if the } i\text{th effect contains } T_{jl}, \\ 0, & \text{if the } i\text{th effect contains } T_{jq}. \end{cases} \tag{1.5.4}$$

Proposition 1.5.1. *Let there are k_{C2} two-level and k_{C3} three-level control factors, k_n noise factors, and k_I three-level factors with internal noise. The loss function defined in (1.2.7) is*

$$L(\boldsymbol{D}) = tr(\boldsymbol{A}var(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{D})),$$

where $\boldsymbol{A}$ is a diagonal matrix, whose diagonal entries are given by

$$A_{ii} = \begin{cases} 1, & \textit{if the effect contains one and only one n, i.e., n, Cn, } T_l n, \ldots \\ \sum_{j=1}^{k_I} i_j 12/8^{\delta_{i,j}}, & \textit{if the effect contains T but no n, i.e., } T_l, T_q, CT_l, CT_q, \ldots \\ 0, & \textit{otherwise.} \end{cases} \tag{1.5.5}$$

Consider again the Example 4. Using Proposition 1.5.1, we obtain the loss function as

$$L(\boldsymbol{D}) = \{\frac{3}{2}V_1 + \frac{3}{2}V_4 + 12V_7 + 12V_8 + V_3 + V_5 + V_6 + V_9 + V_{10} + V_{11}\}.$$

For the frequentist estimation of the effects associated with x_1, z_2, T_{1l}, T_{1q}, x_1z_2, $T_{1l}z_2$, $T_{1l}x_1$, and the grand mean, $m \geq 8$. Therefore, we choose $m = 8$. Using the same definition for $U(\boldsymbol{D})$ as before, we can directly apply our exchange algorithm with $\boldsymbol{D}$ replaced by $\bar{\boldsymbol{D}}$. The optimal design $\bar{\boldsymbol{D}}_1$ is given in Table 1.5.1. Interestingly, the optimal design happened to be

Table 1.5.1: $\bar{\boldsymbol{D}}_1$ and $\bar{\boldsymbol{D}}_2$ for Example 4.

	$\bar{\boldsymbol{D}}_1$			$\bar{\boldsymbol{D}}_2$		
run	x_1	z_2	t_1	x_1	z_2	t_1
1	−1	−1	−1	−1	−1	−1
2	1	1	−1	1	1	−1
3	−1	1	0	−1	1	0
4	1	−1	0	1	−1	0
5	−1	−1	0	−1	−1	1
6	1	1	0	1	1	1
7	−1	1	1	−1	1	1
8	1	−1	1	1	−1	1

Table 1.5.2: Average absolute correlation for $\bar{\boldsymbol{D}}_1$ and $\bar{\boldsymbol{D}}_2$ in Example 4.

Design	x_1	z_2	t_{1l}	$t_{1l}x_1$	$t_{1l}z_2$	x_1z_2	t_{1q}	$t_{1q}x_1$	$t_{1q}z_2$	$t_{1l}x_1z_2$	$t_{1q}x_1z_2$
$\bar{\boldsymbol{D}}_1$	0.102	0.102	0.115	0.115	0.115	0.102	0.100	0.076	0.076	0.100	0.076
$\bar{\boldsymbol{D}}_2$	0.133	0.133	0.141	0.135	0.135	0.141	0.159	0.162	0.162	0.141	0.186

an orthogonal main effects plan $OME(8, 3^1 2^2)$, which can be produced by collapsing the two middle levels of the four-level factor in $OA(8, 4^1 2^2)$ into level 0 (Wu and Hamada 2000, Section 7.8). For comparison, suppose we construct another $OME(8, 3^1 2^2)$ by collapsing the highest two levels of the four-level factor into level 1. This design $\bar{\boldsymbol{D}}_2$ is also shown in Table 1.5.1. It can be shown that $U(\bar{\boldsymbol{D}}_1) > U(\bar{\boldsymbol{D}}_2)$ for all $\rho \in (0, 1)$ and therefore, $\bar{\boldsymbol{D}}_1$ is a better design than $\bar{\boldsymbol{D}}_2$. This could be because $\bar{\boldsymbol{D}}_1$ can estimate the quadratic effect of T_1 better and thus, can obtain a better estimate of the control-by-noise interaction t_1z_1 than $\bar{\boldsymbol{D}}_2$. In Table 1.5.2 we compare the average absolute correlations for each effect in model (1.5.1). Clearly, $\bar{\boldsymbol{D}}_1$ is a better design compared to $\bar{\boldsymbol{D}}_2$ since it has less correlation for every effect.

1.6 Conclusions

In this article we proposed a Bayesian optimal design criterion for constructing single arrays for robust parameter design experiments. Different from others, the Bayesian criterion is capable of incorporating the importance of control-by-noise interactions without altering the effect hierarchy principle. Several examples show the superiority of the proposed single

arrays; in both Bayesian and frequentist viewpoints.

We have made several assumptions to simplify the approach. Specifically, we have chosen to fix $r = 1/3$ and $\sigma^2/\tau^2 = 0$ for finding the optimal single arrays. A better approach could be to postulate a second stage prior on these hyperparameters and give a fully Bayesian treatment, but this will be at the cost of increased computations. Many examples that we tried so far show that the foregoing choices of r and σ^2/τ^2 are quite reasonable. Moreover, in many cases the single arrays we obtained are uniformly optimal for all choices of these hyperparameters.

It is also possible to incorporate effect heredity principle (see, e.g., Wu and Hamada 2000) in our Bayesian optimal design criterion. This can be done using a different ρ_i for each factor. See Joseph and Delaney (2007) for the details of this prior specification. However, rarely those values will be known before the experiment and therefore, a fully Bayesian treatment is warranted. We leave this as a topic for future research.

1.7 Appendix: Proofs

Proof of (1.3.9)

We have:

$$\tilde{\boldsymbol{U}}_{\mathbf{D}}\boldsymbol{R}\tilde{\boldsymbol{U}}_{\mathbf{D}} = \begin{pmatrix} \boldsymbol{U}_{\mathbf{D}} \\ \boldsymbol{F}' \end{pmatrix} \boldsymbol{R}\,(\boldsymbol{U}_{\mathbf{D}}', \boldsymbol{F}) = \begin{pmatrix} \boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}', & \boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{F} \\ \boldsymbol{F}'\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}', & \boldsymbol{F}'\boldsymbol{R}\boldsymbol{F} \end{pmatrix}$$

and thus,

$$(\tilde{\boldsymbol{U}}_{\mathbf{D}}\boldsymbol{R}\tilde{\boldsymbol{U}}_{\mathbf{D}}')^{-1} = \frac{1}{d}\begin{pmatrix} d(\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}')^{-1} + \boldsymbol{H}'\boldsymbol{F}\boldsymbol{F}'\boldsymbol{H}, & -\boldsymbol{H}'\boldsymbol{F} \\ -\boldsymbol{F}'\boldsymbol{H}, & 1 \end{pmatrix},$$

where $d = \boldsymbol{F}'(\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F}$, $\boldsymbol{H} = \boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}'(\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}')^{-1}$, and $\boldsymbol{M} = \boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}'(\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}')^{-1}\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R} = \boldsymbol{H}\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}$. Hence, we obtain (1.3.9):

$$\begin{aligned}\boldsymbol{M}_x &= (\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}', \boldsymbol{R}\boldsymbol{F})\,\frac{1}{d}\begin{pmatrix} d(\boldsymbol{U}_{\mathbf{D}}\boldsymbol{R}\boldsymbol{U}_{\mathbf{D}}')^{-1} + \boldsymbol{H}'\boldsymbol{F}\boldsymbol{F}'\boldsymbol{H}, & -\boldsymbol{H}'\boldsymbol{F} \\ -\boldsymbol{F}'\boldsymbol{H}, & 1 \end{pmatrix}\begin{pmatrix} \boldsymbol{U}_{\mathbf{D}}\boldsymbol{R} \\ \boldsymbol{F}'\boldsymbol{R} \end{pmatrix} \\ &= \boldsymbol{M} + \frac{1}{d}\left((\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F}\right)\left((\boldsymbol{R} - \boldsymbol{M})\boldsymbol{F}\right)'.\end{aligned}$$

Proof of Propositions 1.3.1 and 1.4.1

Because Proposition 1.3.1 is a special case of Proposition 1.4.1 when $k_{C3} = 0$ and $k_{C2} > 0$, we only need to prove Proposition 1.4.1. We can write the transfer function (1.4.1) as

$$\begin{aligned}f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}) = \mu_0 + \sum_{i=0}^{N-1}\beta_i^0 v_i + \left\{\sum_{i=0}^{N-1}\beta_i^1 v_i\right\} z_1 + \cdots + \left\{\sum_{i=0}^{N-1}\beta_i^{k_n} v_i\right\} z_{k_n} \\ + \left\{\sum_{i=0}^{N-1}\beta_i^{12} v_i\right\} z_1 z_2 + \cdots + \left\{\sum_{i=0}^{N-1}\beta_i^{12\ldots k_n} v_i\right\} z_1 \ldots z_{k_n},\end{aligned} \tag{1.7.1}$$

where z_i are noise factors, v_i's are generic notations for the linear, quadratic, and interaction effects among control factors, and $N = 2^{k_{C2}}3^{k_{C3}} = |\mathscr{X}|$. Let x_{kl} and x_{kq} denote the linear and quadratic effects of x_k defined based on the coding in (2.5.3). Note that because the coding is the same for a qualitative factor, for simplicity, we use the same terms "linear" and "quadratic" effects to refer to its two components. Then, we can write v_i in the form

$$v_i = \prod_{k=1}^{k_{C2}} x_k^{\gamma_k} \prod_{k=1}^{k_{C3}} x_{kl}^{\min\{\alpha_{i,k},\gamma_k\}} x_{kq}^{\min\{1-\alpha_{i,k},\gamma_k\}}. \tag{1.7.2}$$

Because of the coding scheme, we obtain

$$\begin{aligned}\text{for a two-level factor:} \quad & \sum_{x_k=-1,1} 1 = \sum_{x_k=-1,1} x_k^2 = 2 \\ \text{for a three-level factor:} \quad & \sum_{x_k=-1,0,1} 1 = \sum_{x_k=-1,0,1} x_{kl}^2 = \sum_{x_k=-1,0,1} x_{kq}^2 = 3.\end{aligned} \tag{1.7.3}$$

Define $\gamma_k = 1$ if v_i contains x_k and 0 otherwise. For a three-level factor x_k, if $\gamma_k = 1$, define $\alpha_{i,k} = 1$ if v_i contains x_{kl} and $\alpha_{i,k} = 0$ if v_i contains x_{kq}. Note that v_i cannot contain both x_{kl} and x_{kq} for the same x_k.

From the transfer function, we obtain

$$d(\boldsymbol{x},\boldsymbol{\beta},\widehat{\boldsymbol{\beta}}) = \left(\sum_{i=0}^{N-1}(\beta_i^1 - \hat{\beta}_i^1)v_i, \ldots, \sum_{i=0}^{N-1}(\beta_i^{k_n} - \hat{\beta}_i^{k_n})v_i\right)'. \tag{1.7.4}$$

Apparently, $\sum_{\boldsymbol{x}\in\mathscr{X}} v_i v_j = 0$ if $i \neq j$ and $\sum_{\boldsymbol{x}\in\mathscr{X}} v_i^2 = 2^{k_{C2}}3^{k_{C3}}$. Therefore,

$$\begin{aligned} L(\boldsymbol{D}) &= \frac{1}{|\mathscr{X}|}\sum_{\boldsymbol{x}\in\mathscr{X}} E\{d(\boldsymbol{x},\boldsymbol{\beta},\widehat{\boldsymbol{\beta}})' d(\boldsymbol{x},\boldsymbol{\beta},\widehat{\boldsymbol{\beta}})\} \\ &= \frac{1}{|\mathscr{X}|}E\{\sum_{\boldsymbol{x}\in\mathscr{X}} d(\boldsymbol{x},\boldsymbol{\beta},\widehat{\boldsymbol{\beta}})' d(\boldsymbol{x},\boldsymbol{\beta},\widehat{\boldsymbol{\beta}})\} \\ &= E\{\sum_{j=1}^{k_n}\sum_{i=0}^{N-1}(\beta_i^j - \hat{\beta}_i^j)^2\} \end{aligned}$$

Because $\hat{\boldsymbol{\beta}} = E(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{D})$, $E\{(\beta_i^j - \hat{\beta}_i^j)^2\} = EE\{(\beta_i^j - \hat{\beta}_i^j)^2|\boldsymbol{y},\boldsymbol{D}\} = E\{var(\beta_i^j|\boldsymbol{y},\boldsymbol{D})\}$. Now, by the property of normal distribution $var(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{D})$ is independent of $\boldsymbol{y}$. Thus, we obtain

$$L(\boldsymbol{D}) = \sum_{j=1}^{k_n}\sum_{i=0}^{N-1} var(\beta_i^j|\boldsymbol{y},\boldsymbol{D}) = tr(\boldsymbol{A}var(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{D})),$$

where $\boldsymbol{A}$ follows the definition in Proposition 1.3.1 and 1.4.1.

Proof of Proposition 1.5.1

Let T_k be a factor with internal noise and t_k its nominal value. Using orthogonal polynomial coding, T_{kl} and T_{kq} are the linear and quadratic effects of T_k. Accordingly, $t_{kl} = \sqrt{3/2}t_k$ and $t_{kq} = 3/\sqrt{2}(t_k^2 - 2/3)$. In this proof, $\mathscr{X}$ is the design space consisting of all the control factors x_k, $k = 1,\ldots,k_{C2}+k_{C3}$, and t_k, $k = 1,\ldots,k_I$. Write the transfer function in the same form as in (1.7.1). Let v_i be the effects involving only control factors and factors with internal noise. We can write v_i as:

$$v_i = \prod_{k=1}^{k_{C2}} x_k^{\gamma_k}\prod_{k=1}^{k_{C3}} x_{kl}^{\min\{\alpha_{i,k},\gamma_k\}} x_{kq}^{\min\{1-\alpha_{i,k},\gamma_k\}}\prod_{k=1}^{k_I} T_{kl}^{\min\{\delta_{i,k},i_k\}} T_{kq}^{\min\{1-\delta_{i,k},i_k\}}, \tag{1.7.5}$$

where γ_k and $\alpha_{i,k}$ are defined in the proof of Proposition 1.4.1 and i_k and $\delta_{i,k}$ are defined in (1.5.3) and (1.5.4). Let $N = 2^{k_{C2}}3^{k_{C3}+k_I} = |\mathscr{X}|$.

Now, let us study an element of $d(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{\beta}, \widehat{\boldsymbol{\beta}})$ (use d for short). When

$$d_j = \frac{\partial f(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{0}, \boldsymbol{\beta})}{\partial z_j} - \frac{\partial f(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{0}, \widehat{\boldsymbol{\beta}})}{\partial z_j} = \sum_{i=0}^{N-1} (\beta_i^j - \hat{\beta}_i^j) v_i,$$

where v_i is the same as in (1.7.5) except that T_{kl} and T_{kq} are replaced by the nominal values t_{kl} and t_{kq} for $k = 1, \ldots, k_I$. Then we have $\sum_{\mathscr{X}} v_i v_j = 0$ and $\sum_{\mathscr{X}} v_i^2 = N$, and hence

$$\frac{1}{|\mathscr{X}|} \sum_{\mathscr{X}} \int d_j^2 p(\boldsymbol{\beta}, \boldsymbol{y}) d\boldsymbol{\beta} d\boldsymbol{y} = \sum_{i=0}^{N-1} var(\beta_i^j | \boldsymbol{y}).$$

When

$$d_j = \frac{\partial f(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{0}, \boldsymbol{\beta})}{\partial T_j} - \frac{\partial f(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{0}, \widehat{\boldsymbol{\beta}})}{\partial T_j} = \sum_{i=0}^{N-1} (\beta_i^0 - \hat{\beta}_i^0) \frac{\partial v_i}{\partial T_j},$$

define

$$b_j = \prod_{k=1}^{k_{C2}} x_k^{\gamma_k} \prod_{k=1}^{k_{C3}} x_{kl}^{\min\{\alpha_{i,k}, \gamma_k\}} x_{kq}^{\min\{1-\alpha_{i,k}, \gamma_k\}} \prod_{k \neq j}^{k_I} t_{kl}^{\min\{\delta_{i,k}, i_k\}} t_{kq}^{\min\{1-\delta_{i,k}, i_k\}}.$$

Note that $\sum_{\mathscr{X}} b_j^2 = \sum_{\mathscr{X}} b_j^2 t_{jl}^2 = N$. There are only three cases for v_i:

1. If i_j for T_j is 0, then $\frac{\partial v_i}{\partial T_j} = 0$.
2. If $i_j = 1$ and $\delta_{i,j} = 1$, then $\frac{\partial v_i}{\partial T_j} = \sqrt{\frac{3}{2}} b_j$. Therefore, $\sum_{\mathscr{X}} \left(\frac{\partial v_i}{\partial T_j}\right)^2 = \frac{3}{2} N$.
3. If $i_j = 1$ and $\delta_{i,j} = 0$, then $\frac{\partial v_i}{\partial T_j} = 3\sqrt{2} t_j b_j = 2\sqrt{3} t_{jl} b_j$. Therefore, $\sum_{\mathscr{X}} \left(\frac{\partial v_i}{\partial T_j}\right)^2 = 12N$.

It is easy to see that in all the three cases $\sum_{\mathscr{X}} \frac{\partial v_i}{\partial T_j} \frac{\partial v_{i'}}{\partial T_j} = 0$ if $i \neq i'$. Hence

$$\begin{aligned} \frac{1}{|\mathscr{X}|} \sum_{\mathscr{X}} \int d_j^2 p(\boldsymbol{\beta}, \boldsymbol{y} | \boldsymbol{D}) d\boldsymbol{\beta} d\boldsymbol{y} &= \sum_{i=0}^{N-1} \left\{ i_j \left(\frac{3}{2}\right)^{\delta_{i,j}} \times 12^{1-\delta_{i,j}} \right\} var(\beta_i^0 | \boldsymbol{y}, \boldsymbol{D}) \\ &= \sum_{i=0}^{N-1} \left\{ 12 i_j 8^{-\delta_{i,j}} \right\} var(\beta_i^0 | \boldsymbol{y}, \boldsymbol{D}). \end{aligned}$$

At last, the loss function is computed as follows:

$$\begin{aligned} L(\boldsymbol{D}) &= \frac{1}{|\mathscr{X}|} \sum_{j=1}^{k_I} \sum_{\mathscr{X}} \int d_j^2 p(\boldsymbol{\beta}, \boldsymbol{y} | \boldsymbol{D}) d\boldsymbol{\beta} d\boldsymbol{y} \\ &= \sum_{i=0}^{N-1} \left\{ \sum_{j=1}^{k_I} 12 i_j 8^{-\delta_{i,j}} \right\} var(\beta_i^0 | \boldsymbol{y}, \boldsymbol{D}) + \sum_{j=1}^{k_n} \sum_{i=0}^{N-1} var(\beta_i^j | \boldsymbol{y}, \boldsymbol{D}) \\ &= tr(\boldsymbol{A} var(\boldsymbol{\beta} | \boldsymbol{y}, \boldsymbol{D})). \end{aligned}$$

Thus, the diagonal entries of $\boldsymbol{A}$ corresponding to β_i^0 is $\sum_{j=1}^{k_I} 12 i_j 8^{-\delta_{i,j}}$, corresponding to β_i^j is 1, and all others are 0.

CHAPTER II

BAYESIAN OPTIMAL BLOCKING OF FACTORIAL DESIGNS

2.1 Introduction

Blocking is a fundamental technique in design of experiments. It helps in improving the estimation of treatment effects by eliminating or reducing some of the known sources of variation in the experiment. Because there are two wordlength patterns in blocked fractional factorial designs, i.e., treatment and block wordlength patterns, many researchers have focused on combining them into one wordlength pattern to search for optimal blocked designs. The recent work includes Bisgaard (1994), Sun, Wu and Chen (1997), Sitter, Chen and Feder (1997), Chen and Cheng (1999), Zhang and Park (2000), Cheng and Wu (2002), Ai and Zhang (2004), Xu (2006), and Xu and Lau (2006). However, ambiguities exist in combining the treatment and block wordlength patterns as evidenced by numerous proposals with none emerging as the "best". Moreover, the applicability of these methods is limited in terms of number of runs and factor levels. In addition, there is no easy way to distinguish between qualitative and quantitative factors.

Here we propose a Bayesian approach to overcome the foregoing limitations with the existing methods. The idea is to postulate a model and a prior distribution for the treatment and block effects that satisfy the common assumptions in blocking and then to develop an optimal design criterion for the efficient estimation of treatment effects. Recently, Kang and Joseph (2009) used a similar idea for developing single arrays for robust parameter design. However, the objectives of blocking are different from that of the robustness studies and therefore, the optimal design criterion and the resulting blocked designs are quite different from the single arrays.

This chapter is arranged as follows. In Section 2.2, we briefly review the existing optimality criteria for blocked fractional factorial designs. In Section 2.3, a Bayesian framework for the optimal selection of blocked designs is established and a unified criterion is developed for both regular and nonregular blocked mixed-level factorial designs. We apply the proposed method to develop two-level designs in Section 2.4 and mixed-level designs in Section 2.5. Some concluding remarks are given in Section 2.6.

2.2 Review of the Existing Optimality Criteria

A regular 2^{p-k} design D with 2^{p-k} runs and p factors, denoted by $1, 2, \ldots, p$, is uniquely determined by k independent defining words, say $w_1, \ldots, w_k$. The group formed by the k defining words is represented by $G_t = \{I, w_1, \ldots, w_{2^k-1}\}$ is called the treatment defining contrast subgroup. Let $A_i^t(D)$ be the number of words of length i in G_t, and the vector $W_t(D) = (A_1^t(D), A_2^t(D), A_3^t(D), \ldots, A_p^t(D))$ is called the *treatment wordlength pattern*. Note that the original definition requires $A_1^t = 0$ and $A_2^t = 0$, but here we ignore this restriction so that it can be applied to more general cases.

Arranging a regular 2^{p-k} design into 2^m blocks of size 2^{p-k-m} ($m < p - k$), denoted by $(2^{p-k} : 2^m)$, is equivalent to selecting m independent columns $v_1, \ldots, v_m$ for the m blocking factors $b_1, \ldots, b_m$, i.e., letting $b_1 = v_1, \ldots, b_m = v_m$, which are called block defining relations. The group formed by the $k + m$ treatment and block defining words $\{w_1, \ldots, w_k, b_1v_1, \ldots, b_mv_m\}$ is denoted by G_{t+b}. Then $G_{t+b} \backslash G_t$, denoted by $G_{b\otimes t}$, is the set consisting of all treatment effects confounded with block effects. Let $A_i^b(D)$ be the number of words containing i treatment factors in $G_{b\otimes t}$. The vector $W_b(D) = (A_1^b(D), A_2^b(D), \ldots, A_p^b(D))$ is called the *block wordlength pattern*. Note that as in the case of treatment wordlength pattern, here we include $A_1^b(D)$ in the block wordlength pattern.

For example, consider a blocked $(2^{6-2} : 2^2)$ design D in which the four independent factors are denoted by $1, 2, 3$ and 4. The two additional factors are defined by the treatment defining relations $5 = 123$ and $6 = 234$, and the block defining relations are $b_1 = 134$ and

$b_2 = 124$. The complete defining relation is presented as follows:

$$\begin{aligned} I &= 1235 = 2346 = 1456 \\ &= 134b_1 = 245b_1 = 126b_1 = 356b_1 \\ &= 124b_2 = 345b_2 = 136b_2 = 256b_2 \\ &= 23b_1b_2 = 15b_1b_2 = 46b_1b_2 = 123456b_1b_2. \end{aligned}$$

In the above defining relation, G_t consists of the words in the first row, whereas $G_{b\otimes t}$ consists of all of the remaining words. The treatment and block wordlength patterns are $W_t(D) = (0, 0, 0, 3, 0, 0)$ and $W_b(D) = (0, 3, 8, 0, 0, 1)$.

Zhang and Park (2000) showed that there is no minimum aberration (MA) design with respect to both treatment and block wordlength patterns. To overcome this problem, Sitter, Chen and Feder (1997) (abbreviated as SCF) proposed the following combined wordlength pattern

$$W_{SCF}(D) = (A_1^t(D), A_2^t(D), A_1^b(D), A_3^t(D), A_2^b(D), A_4^t(D), A_3^b(D), \ldots). \tag{2.2.1}$$

Subsequently, from the view-point of estimation capacity, Chen and Cheng (1999) (CC) introduced another combined wordlength pattern:

$$W_{CC}(D) = (A_1^t(D) + A_1^b(D), A_2^t(D), 3A_3^t(D) + A_2^b(D), A_4^t(D), 10A_5^t(D) + A_3^b(D), \ldots), \tag{2.2.2}$$

whereas Cheng and Wu (2002) proposed two other combined wordlength patterns:

$$W_1(D) = (A_1^t(D), A_2^t(D), A_1^b(D), A_3^t(D), A_4^t(D), A_2^b(D), A_5^t(D), A_6^t(D), A_3^b(D), \ldots) \tag{2.2.3}$$

$$W_2(D) = (A_1^t(D), A_1^b(D), A_2^t(D), A_3^t(D), A_2^b(D), A_4^t(D), A_5^t(D), A_3^b(D), A_6^t(D), \ldots) \tag{2.2.4}$$

MA criterion can be applied on these combined wordlength patterns for selecting optimal blocked designs. However, it is not clear which among the four wordlength patterns will produce the "best" designs. It is not even clear if such modified orderings are meaningful as they violate the effect hierarchy principle. These ambiguities motivate our research on blocked designs.

A similar approach can be adopted for the case of blocked nonregular designs. Ai and Zhang (2004) extended the definitions of treatment and block wordlength patterns of regular designs to general nonregular designs. Let $Q_s = \{0, 1, \ldots, s-1\}$ be the integer ring with modulus s. An asymmetrical (or mixed-level) design of n runs and p factors with $s_1, \ldots, s_p$ levels, denoted by $(n, s_1 \cdots s_p)$, is a set of n row vectors (or points) in $Q_{s_1} \times \cdots \times Q_{s_p}$ or an $n \times p$ matrix in which each row represents a run, each column represents a factor and the jth column takes values from a set of s_j symbols, say, Q_{s_j}. In particular, an (n, s^p)-design is symmetrical.

An asymmetrical blocked design of n runs, p factors with levels $s_1, \ldots, s_p$ in $(s_{p+1} \cdots s_{p+m})$ blocks formed by the level combinations of m factors with levels $s_{p+1}, \ldots, s_{p+r}$, denoted by $(n, s_1 \cdots s_p : s_{p+1} \cdots s_{p+m})$, is an $(n, s_1 \cdots s_{p+m})$-design in which the first p factors are treatment factors and the remaining m factors are blocking factors. For an s-level factor, let $\chi_v(z)$ be the orthonormal polynomial contrast coefficient of level z for $v \in Q_s$ satisfying $\sum_{z \in Q_s} \chi_{v_1}(z)\chi_{v_2}(z) = s\delta_{v_1,v_2}$ for any $v_1, v_2 \in Q_s$, where $\delta_{v_1,v_2} = 1$ when $v_1 = v_2$ and 0 otherwise. Let $\chi_0(z) = 1$ for any $z \in Q_s$. For example, for a two-level factor, the two orthonormal contrast coefficient vectors are $(1, 1)$ and $(-1, 1)$, while for a three-level factor, the three orthonormal contrast coefficient vectors are $(1, 1, 1)$, $(-\sqrt{6}/2, 0, \sqrt{6}/2)$ and $(\sqrt{2}/2, -\sqrt{2}, \sqrt{2}/2)$.

For a blocked $(n, s_1 \cdots s_p : s_{p+1} \cdots s_{p+m})$-design $D = (d_{ij}) = (\boldsymbol{d}_1', \ldots, \boldsymbol{d}_n')'$, let $Q_t = Q_{s_1} \times \cdots \times Q_{s_p}$ and $Q_b = Q_{s_{p+1}} \times \cdots \times Q_{s_{p+m}}$. For any $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2)$ with $\boldsymbol{v}_1 = (v_1, \ldots, v_p) \in Q_t$ and $\boldsymbol{v}_2 = (v_{p+1}, \ldots, v_{p+m}) \in Q_b$, define

$$J_{\mathbf{v}}(D) = \sum_{i=1}^{n} \chi_{\mathbf{v}}(\boldsymbol{d}_i), \quad \text{and } \chi_{\mathbf{v}}(\boldsymbol{d}_i) = \prod_{j=1}^{p+m} \chi_{v_j}(d_{ij}).$$

For $i = 1, \ldots, p$, let

$$\begin{aligned} A_i^t(D) &= n^{-2} \sum_{wt(\mathbf{v}_1)=i, \mathbf{v}_2=\mathbf{0}} [J_{\mathbf{v}}(D)]^2, \\ A_i^b(D) &= n^{-2} \sum_{wt(\mathbf{v}_1)=i, wt(\mathbf{v}_2)\geq 1} [J_{\mathbf{v}}(D)]^2, \end{aligned}$$

where $wt(\boldsymbol{v}_1)$ is the number of nonzero elements of a vector $\boldsymbol{v}_1$ and $\mathbf{0}$ is a vector of 0's. The vectors $W_t(D) = (A_1^t(D), A_2^t(D), \ldots, A_p^t(D))$ and $W_b(D) = (A_1^b(D), A_2^b(D), \ldots, A_p^b(D))$ are the treatment and block *generalized wordlength patterns*, respectively. The previous four combined wordlength patterns W_{SCF}, W_{CC}, W_1 and W_2 can be obtained by correspondingly ordering the components of the generalized wordlength patterns W_t and W_b into one vector.

The ambiguities present in the wordlength patterns of blocked regular designs carry over to the case of nonregular blocked designs as well. Moreover, the number of runs are restricted to be a power of 2 or multiple of 4 and therefore, the existing methods are not flexible enough to find optimal designs for any number of runs. Furthermore, there is no easy way to deal with qualitative and quantitative factors. We propose a Bayesian approach to overcome these limitations.

2.3 Bayesian Optimal Criterion for Blocking Schemes

The objective is to design an efficient experiment so as to estimate the treatment effects precisely. Although the block effects are not of any direct interest to the experimenter, they can be quite significant and therefore, the block effects cannot be ignored in estimation. We formulate a Bayesian optimal design criterion to satisfy these objectives. For the model and prior specification, we follow the usual assumptions (see, e.g., Cheng and Wu, 2002).

Assumptions:

(i) Interactions between block factors and treatment factors are negligible.

(ii) *Effect hierarchy principle*: Lower-order treatment effects are more likely to be significant than higher-order treatment effects and the treatment effects of the same order are equally likely to be significant.

(iii) Block effects are more likely to be significant than treatment effects.

(iv) Interactions between two or more block factors have the same importance as the main effects of block factors.

Suppose that the output Y is related to the treatment factors $\boldsymbol{x}_1 = (x_1, \ldots, x_p)'$ and blocking factors $\boldsymbol{x}_2 = (x_{p+1}, \ldots, x_{p+m})'$ by the model $Y = f(\mathbf{x}) + e$, where $\boldsymbol{x} = (\boldsymbol{x}_1', \boldsymbol{x}_2')'$ and $e \sim N(0, \sigma^2)$ is the random error in the output. We approximate $f(\mathbf{x})$ by a linear model. By assumption (i), we can ignore the interactions between block and treatment effects. Thus, let

$$f(\mathbf{x}) = \mu_0 + \sum_{\mathbf{v}_1 \in Q_t} \chi_{\mathbf{v}_1}(\boldsymbol{x}_1)\beta_{\mathbf{v}_1} + \sum_{\mathbf{v}_2 \in Q_b^0} \chi_{\mathbf{v}_2}(\boldsymbol{x}_2)\beta_{\mathbf{v}_2}, \tag{2.3.1}$$

where μ_0 is a constant, $Q_b^0 = Q_b \setminus \{\mathbf{0}\}$, $\boldsymbol{\beta}_t = (\beta_{\mathbf{v}_1}, \boldsymbol{v}_1 \in Q_t)'$ represent the $\prod_{i=1}^{p} s_i$ treatment effect components consisting of the gross mean $\beta_{\mathbf{0}}$, $(\sum_{i=1}^{p} s_i - p)$ main effect components (m.e.), $\sum_{i<j}(s_i - 1)(s_j - 1)$ two-factor interaction components (2fi), ..., and $\prod_{i=1}^{p}(s_i - 1)$ p-factor interaction components, while $\boldsymbol{\beta}_b = (\beta_{\mathbf{v}_2}, \boldsymbol{v}_2 \in Q_b^0)'$ represent the $(\prod_{i=p+1}^{p+m} s_i - 1)$ block effects, and $\chi_{\mathbf{v}_1}(\boldsymbol{x}_1)$ and $\chi_{\mathbf{v}_2}(\boldsymbol{x}_2)$ are, respectively, the corresponding orthonormal contrast coefficients. Note that the foregoing model contains no treatment-by-block interaction terms.

Now that assumption (i) is already incorporated through model specification, we are left with the three assumptions (ii)-(iv). They will be incorporated in the model through a Bayesian framework. For a blocked $(n, s_1 \cdots s_p : s_{p+1} \cdots s_{p+m})$ design D, let $\mathbf{y} = (y_1, \ldots, y_n)'$ be the response values obtained from the n runs. Assume that the e's are independent between different runs and σ^2 is known. Assume the prior distribution for $\boldsymbol{\beta} = (\boldsymbol{\beta}_t', \boldsymbol{\beta}_b')'$ to be $N(\mathbf{0}, \tau^2\boldsymbol{R})$. We carefully choose the variance-covariance matrix $\tau^2\boldsymbol{R}$ to reflect the three assumptions (ii)-(iv), which can be easily done using the functionally induced priors in Joseph (2006) and Joseph and Delaney (2007). This will be explained later. Let $\boldsymbol{U}_D = (\boldsymbol{U}_t, \boldsymbol{U}_b)$ be the model matrix whose entries are the orthonormal contrast coefficients corresponding to $\boldsymbol{\beta}$ generated from the blocked design D. Denote $\mathbf{1}$ to be a vector of 1's and $\boldsymbol{I}$ to be the

identity matrix. Thus, the model is

$$\mathbf{y}|\boldsymbol{\beta} \sim N(\mu_0\mathbf{1} + \boldsymbol{U}_D\boldsymbol{\beta}, \sigma^2\boldsymbol{I}), \quad \boldsymbol{\beta} \sim N(\mathbf{0}, \tau^2\boldsymbol{R}). \tag{2.3.2}$$

The objective of the experiment is to estimate the treatment effects precisely. Thus, a good design of experiment should make the posterior variances of the treatment effects' estimates as small as possible. Therefore, we propose to minimize the sum of the posterior variances of the treatment effects, i.e.,

$$\min_D \ \mathrm{tr}(\mathrm{var}(\boldsymbol{\beta}_t|\boldsymbol{y})).$$

The posterior variance of $\boldsymbol{\beta}$ is given by

$$\mathrm{var}(\boldsymbol{\beta}|\boldsymbol{y}) \quad = \quad \tau^2\boldsymbol{R} - \tau^4\boldsymbol{R}\boldsymbol{U}_D'(\tau^2\boldsymbol{U}_D\boldsymbol{R}\boldsymbol{U}_D' + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{U}_D\boldsymbol{R}. \tag{2.3.3}$$

Let $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{R}_t, \boldsymbol{R}_b)$, which implies that the treatment effects $\boldsymbol{\beta}_t$ and block effects $\boldsymbol{\beta}_b$ are assumed to be uncorrelated. Then the posterior variance of $\boldsymbol{\beta}_t$ can be obtained as

$$\begin{aligned}\mathrm{var}(\boldsymbol{\beta}_t|\boldsymbol{y}) \quad &= \quad \tau^2\boldsymbol{R}_t - \tau^4\boldsymbol{R}_t\boldsymbol{U}_t'(\tau^2\boldsymbol{U}_D\boldsymbol{R}\boldsymbol{U}_D' + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{U}_t\boldsymbol{R}_t \\ &= \quad \tau^2\boldsymbol{R}_t - \tau^2\boldsymbol{R}_t\boldsymbol{U}_t'(\boldsymbol{U}_t\boldsymbol{R}_t\boldsymbol{U}_t' + \boldsymbol{U}_b\boldsymbol{R}_b\boldsymbol{U}_b' + \lambda\boldsymbol{I})^{-1}\boldsymbol{U}_t\boldsymbol{R}_t, \end{aligned} \tag{2.3.4}$$

where $\lambda = \sigma^2/\tau^2$. Since $\mathrm{tr}(\boldsymbol{R}_t)$ is a constant independent of design D, minimizing $\mathrm{tr}(\mathrm{var}(\boldsymbol{\beta}_t|\boldsymbol{y}))$ reduces to maximizing

$$B(D) = \mathrm{tr}[\boldsymbol{R}_t\boldsymbol{U}_t'(\boldsymbol{U}_t\boldsymbol{R}_t\boldsymbol{U}_t' + \boldsymbol{U}_b\boldsymbol{R}_b\boldsymbol{U}_b' + \lambda\boldsymbol{I})^{-1}\boldsymbol{U}_t\boldsymbol{R}_t]. \tag{2.3.5}$$

This motivates us to define a new Bayesian optimal criterion. A blocked $(n, s_1 \cdots s_p : s_{p+1} \cdots s_{p+m})$ design D is called *Bayesian-optimal* or *B-optimal* if it maximizes $B(D)$.

The foregoing criterion is very general and can be used for factors with any number of levels and places no restrictions on the number of runs, type of designs, or type of factors. We illustrate the approach, first using two-level designs and then, using mixed two- and three-level designs.

2.4 Blocking of Two-Level Factorial Designs

We use the functionally induced priors in Joseph (2006) for the prior specification of the treatment and block effects. For the treatment effects, let $\boldsymbol{\beta}_t \sim N(\mathbf{0}, \tau^2 \boldsymbol{R}_t)$, where

$$\boldsymbol{R}_t = \text{diag}(1, r, \ldots, r, r^2, \ldots, r^2, \ldots, r^p)$$

and r is a value between 0 and 1. Therefore, the parameters $\boldsymbol{\beta}_t$ are independent with variances $\text{var}(\beta_0) = \tau^2$, $\text{var}(\beta_{m.e.}) = \tau^2 r$, $\text{var}(\beta_{2fi}) = \tau^2 r^2$, etc. When $r \in (0, 1)$, the variances decrease geometrically as the order of the effects increase, thus satisfying effect hierarchy assumption in (ii).

A similar prior distribution can be used for the block effects as well. Let $\boldsymbol{\beta}_b \sim N(\mathbf{0}, \tau^2 \boldsymbol{R}_b)$, where $\boldsymbol{R}_b = \text{diag}(r_b, \ldots, r_b^2, \ldots, r_b^m)$ and $r_b \in [0, 1]$. By assumption (iv), we must have $r_b = 1$. Thus, $\boldsymbol{R}_b = \boldsymbol{I}$ and $\boldsymbol{\beta}_b \sim N(\mathbf{0}, \tau^2 \boldsymbol{I})$. Note that the variance of a block effect (τ^2) will always be more than the variance of a treatment effect ($\tau^2 r^k$); thus satisfying assumption (iii).

Now the B-optimal design can be obtained by maximizing $B(D)$ in (2.3.5). Of course, we need to specify the values of r and λ for finding the B-optimal design. Based on a study conducted by Li, Sudarsanam, and Frey (2006) on 113 full factorial experiments, Kang and Joseph (2009) argued that $r = 1/3$ is a good choice in the absence of any other prior knowledge about the process. Also, by assuming a high signal-to-ratio we can neglect the ratio $\lambda = \sigma^2/\tau^2$. Then, the B-optimal design criterion reduces to maximizing

$$B(D) = \text{tr}[\boldsymbol{R}_t \boldsymbol{U}_t' (\boldsymbol{U}_t \boldsymbol{R}_t \boldsymbol{U}_t' + \boldsymbol{U}_b \boldsymbol{U}_b')^{-1} \boldsymbol{U}_t \boldsymbol{R}_t], \tag{2.4.1}$$

with $r = 1/3$.

We use a modified exchange algorithm proposed in Kang and Joseph (2009) for searching the optimal design. The computer code is available from the authors upon request. Because this algorithm can be easily used for generating optimal designs for any number of factors or runs, we do not tabulate the optimal designs.

2.4.1 Simplification for Regular Two-Level Designs

The optimal design criterion can be greatly simplified for the case of regular designs. This helps in computation and can also provide more insights about the criterion. Since in a regular $(2^{p-k} : 2^m)$ design D, any two effects are either fully aliased or independent, all the 2^p treatment effects can be divided into 2^{p-k} mutually exclusive aliasing sets of 2^k effects, each being a coset of G, and among which the $2^m - 1$ sets are fully confounded with block effects. Let $G_0 = G$ for the convenience of later citation, $G_1, \ldots, G_{2^m-1}$ be the aliasing sets confounded with a block effect, and $G_{2^m}, \ldots, G_{2^{p-k}-1}$ be the remaining aliasing sets. Correspondingly, reorder $\boldsymbol{\beta}$ by putting together the effects belonging to the same aliasing set as $\boldsymbol{\beta} = (\boldsymbol{\beta}^{(0)'}, \boldsymbol{\beta}^{(1)'}, \ldots, \boldsymbol{\beta}^{(2^{p-k}-1)'})'$, where $\boldsymbol{\beta}^{(0)} = (\beta_0^{(0)}, \beta_1^{(0)}, \ldots, \beta_{2^k-1}^{(0)})'$ represent the gross mean ($\beta_0^{(0)} = \beta_0$) and all the treatment effects aliased with it in G_0, and for $j = 1, \ldots, 2^m - 1$, $\boldsymbol{\beta}^{(j)} = (\beta_0^{(j)}, \beta_1^{(j)}, \ldots, \beta_{2^k}^{(j)})'$ represents the block effect ($\beta_0^{(j)} = \beta_{bj}$) and the 2^k treatment effects in G_j. All other $\boldsymbol{\beta}^{(j)} = (\beta_1^{(j)}, \ldots, \beta_{2^k}^{(j)})'$'s are neither aliased with the gross mean nor confounded with any block effect.

Let $\text{var}(\beta_i^{(j)}) = \tau^2 R_i^{(j)}$ and $B_j = \sum_i R_i^{(j)}$, where the summation is over all possible i for a fixed j. Then, as in Joseph (2006), we obtain

$$\text{var}(\beta_i^{(j)}|\mathbf{y}) \quad = \quad \tau^2 R_i^{(j)} - \tau^2 (R_i^{(j)})^2 \left(\lambda 2^{-t} + B_j\right)^{-1},$$

for $i = 0, 1, \ldots, 2^k$ and $j = 0, 1, \ldots, 2^{p-k} - 1$. Thus, (2.4.1) simplifies to

$$B(D) = \sum_{j=0}^{2^{p-k}-1} \sum_{i=1}^{2^k} (R_i^{(j)})^2 / B_j.$$

For a blocked $(2^{p-k} : 2^m)$ design D, let $N_i^{(j)}$ be the number of treatment effects of length i in G_j for $i = 1, \ldots, p$. Define $N_0^{(j)} = 1$ for $0 \le j \le 2^m - 1$, and $N_0^{(j)} = 0$ for $2^m \le j \le 2^{p-k} - 1$. Denote by N the $2^{p-k} \times (p+1)$ matrix with $N_i^{(j)}$ being the $(j+1, i+1)$-th element, which is called the coset pattern matrix in Zhu and Zeng (2005). Note that according to the previous notations of treatment and block wordlength patterns, $A_i^t = N_i^{(0)}$ and $A_i^b = \sum_{j=1}^{2^m-1} N_i^{(j)}$. Then we have $B_j = \sum_i R_i^{(j)} = \sum_{i=0}^{p} r^i N_i^{(j)}$, for $j = 0, \ldots, 2^{p-k} - 1$. For notational convenience, let

$R_{2^k}^{(0)} = 0$. Then,

$$B(D) = \sum_{j=0}^{2^{p-k}-1} \sum_{i=1}^{2^k} \frac{(R_i^{(j)})^2}{B_j} = \sum_{j=0}^{2^{p-k}-1} \frac{\sum_{i=1}^{p} r^{2i} N_i^{(j)}}{\sum_{i=0}^{p} r^i N_i^{(j)}}. \tag{2.4.2}$$

2.4.2 Examples

In this section, we present two examples of B-optimal blocked designs and compare them with the existing designs. The first example is on regular designs and the second one is on non-regular designs.

Example 1. Consider the following two blocked $(2^{5-1} : 2^1)$ designs D_1 and D_2, whose treatment and block defining generators are:

$$D_1 : 5 = 1234, b_1 = 12; \qquad D_2 : 5 = 123, b_1 = 124.$$

Their complete defining contrast subgroups are $D_1 : I = 12345 = 12b_1 = 345b_1$ and $D_2 : I = 1235 = 124b_1 = 345b_1$. According to the previous four combined wordlength patterns, it can be easily shown that D_1 has less aberration than D_2 under W_1 and D_2 has less aberration than D_1 under W_{SCF}, W_2 and W_{CC}. In fact, by complete search it can be shown that D_1 has MA under W_1 and D_2 has MA under W_{SCF}, W_2 and W_{CC}.

Using (2.4.2), the B values of designs D_1 and D_2 can be obtained as

$$\begin{aligned}
B(D_1) &= \frac{r^{10}}{1+r^5} + \frac{r^4+r^6}{1+r^2+r^3} + 5\frac{r+r^7}{1+r^3} + 9\frac{r^2+r^4}{1+r} \\
&\simeq 5r + 9r^2 - 9r^3 + 14r^4 - 18r^5, \\
B(D_2) &= \frac{r+r^8+r^9}{1+r^4} + \frac{2r^6}{1+2r^3} + \frac{4(r+r^5)+3(r^2+r^6)}{1+r^2} + 3r^2 + 3r^3 \\
&\simeq 5r + 6r^2 - r^3 - 3r^4 + 7r^5.
\end{aligned}$$

The approximate expressions show that $B(D_1)$ is larger than $B(D_2)$ for small values of r. In fact, by numerical computation it can be shown that $B(D_1) > B(D_2)$ for all $r \in (0, 1)$ and therefore, D_1 is uniformly better than D_2. Moreover, at $r = 1/3$, D_1 attains the maximum

Table 2.4.1: Average absolute correlations of the effects of D_1 and D_2

Design	Main effects	Two-factor interactions	Block effects
D_1	0	0.0067	0.0667
D_2	0	0.0400	0

value among all $(2^{5-1} : 2^1)$ designs and consequently is the B-optimal design. Thus, in this example B criterion agrees with W_1, but disagrees with W_{SCF}, W_2, and W_{CC}. To understand which design is really the best, we study these two designs more carefully using some other frequentist measures.

It can be shown that all the five main effects are clear for both the designs, but D_1 has nine clear 2fi's and D_2 has only four. This indicates that D_1 is better than D_2, which agrees with the B-optimal design. We propose another frequentist measure for judging the "goodness" of a design that is useful for comparing nonregular designs as well. Consider three groups of effects: block effects, treatment main effects, and treatment two-factor interactions. Let ρ_{ij} denote the pairwise correlation between ith and jth effects. Then, $\sum_{j\neq i} |\rho_{ij}|/16$ is the average absolute correlation for the ith effect with the other effects. A good blocked design should make this average correlation for the treatment effects as small as possible. Table 2.4.1 shows the average absolute correlations averaged over the effects in each of the three groups. We can see that treatment main effects are uncorrelated with the other effects in both the designs, but D_1 has smaller average absolute correlation for the treatment two-factor interactions than that of D_2. This shows that the treatment two-factor interactions are less contaminated by the other effects in D_1 than D_2. However, D_2 is better for estimating the block effects than D_1; but this does not help the experimenter because he/she does not have any interest in the block effects. Clearly, D_1 has sacrificed the precision of block effects for more efficient estimation of the treatment effects. This is the objective of blocking and this is why D_1 is a better blocked design than D_2.

Example 2. Consider blocked $(12, 2^6 : 2)$ designs with 12 runs in 2 blocks. The B-optimal

Table 2.4.2: B-optimal design D_3 and Placket-Burman design D_4 in Example 2

	D_3							D_4						
Run	t_1	t_2	t_3	t_4	t_5	t_6	b_1	t_1	t_2	t_3	t_4	t_5	t_6	b_1
1	−1	1	−1	−1	−1	−1	−1	1	1	−1	1	1	1	−1
2	1	−1	−1	1	−1	−1	−1	−1	1	1	−1	1	1	1
3	1	−1	1	−1	1	−1	−1	1	−1	1	1	−1	1	1
4	−1	1	−1	1	−1	1	−1	−1	1	−1	1	1	−1	1
5	−1	1	1	−1	1	1	−1	−1	−1	1	−1	1	1	−1
6	1	−1	1	1	1	1	−1	−1	−1	−1	1	−1	1	1
7	−1	−1	1	1	−1	−1	1	1	−1	−1	−1	1	−1	1
8	1	1	1	1	−1	−1	1	1	1	−1	−1	−1	1	−1
9	−1	1	−1	1	1	−1	1	1	1	1	−1	−1	−1	1
10	1	−1	1	−1	−1	1	1	−1	1	1	1	−1	−1	−1
11	−1	−1	−1	−1	1	1	1	1	−1	1	1	1	−1	−1
12	1	1	−1	−1	1	1	1	−1	−1	−1	−1	−1	−1	−1

design D_3 obtained with $r = 1/3$ is given in Table 2.4.2, where the two levels are coded by ± 1. Note that D_3 is balanced, but is not orthogonal. For comparison, an orthogonal design D_4 using the first seven columns of a 12-run Placket-Burman design is also presented in this table.

The relative efficiency $B(D_4)/B(D_3)$ is plotted in Figure 2.4.1, which shows that D_3 is uniformly better than D_4 for all $r \in (0, 1)$. Their generalized wordlength patterns are shown in Table 2.4.3. The four combined wordlength patterns W_{SCF}, W_1, W_2, and W_{CC} can now be computed. They all identify D_4 to be better than D_3. Thus, in this example, the B-optimality criterion does not agree with any of the existing criteria.

Here the maximum clear two-factor interaction criterion cannot be used for comparison. Instead, we use the average absolute correlation criterion. The results are shown in Table 2.4.4. Clearly, D_3 has much less correlation for the main effects compared to D_4 and therefore, D_3 is capable of estimating the treatment main effects more precisely than that of D_4. Although the correlation for two-factor interactions is slightly larger, D_3 is the clear winner. This example shows the superiority of the proposed B-optimal design criterion over the existing criteria.

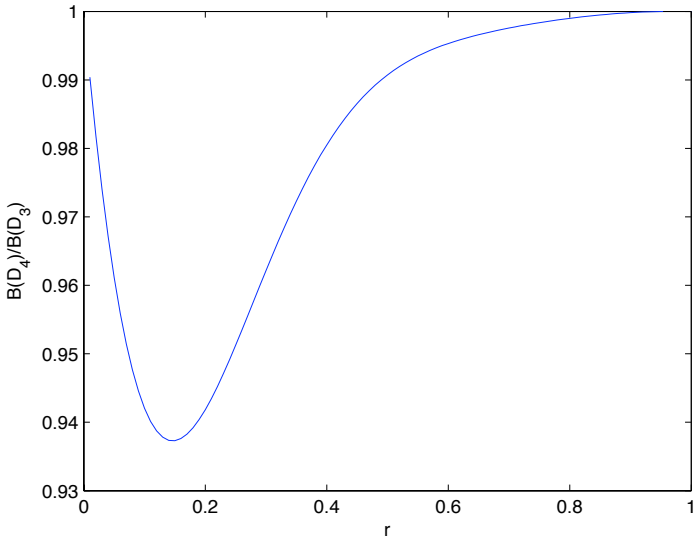


Figure 2.4.1: Relative efficiency: $B(D_4)/B(D_3)$.

Table 2.4.3: Generalized wordlength patterns of D_3 and D_4

D_3	W_t	0	0.6667	0	3.6667	0	0
	W_b	0	2.7778	0	2.4444	0	0.1111
D_4	W_t	0	0	2.2222	1.6667	0.4444	0
	W_b	0	1.6667	2.2222	0.8889	0.4444	0.1111

Table 2.4.4: Average absolute correlations of the effects of D_3 and D_4

Design	Main effects	Two-factor interactions	Block effects
D_3	0.0317	0.2112	0.2419
D_4	0.1587	0.1746	0.2381

2.5 Blocking of Mixed Two- and Three-Level Designs

Experiments with mixed two- and three-level factors are common in practice. However, the issue of optimal blocking in mixed-level experiments has attracted little attention except for the work of Ai and Zhang (2004). In this section we explain how to design such blocked experiments by using the proposed Bayesian approach.

For prior specification, we again use the functionally induced priors in Joseph (2006) and Joseph and Delaney (2007). The prior variance-covariance matrix of $\boldsymbol{\beta}_t$ is given by

$$\text{var}(\boldsymbol{\beta}_t) = \tau^2 \boldsymbol{R}_t = \sigma_0^2 \bigotimes_{j=1}^{p} \boldsymbol{U}_j^{-1} \boldsymbol{\Psi}_j (\boldsymbol{U}_j^{-1})' \tag{2.5.1}$$

where $\boldsymbol{U}_j$ and $\boldsymbol{\Psi}_j$ are the orthonormal contrast coefficient matrix and correlation matrix of the jth factor, σ_0^2 is the prior variance of the underlying transfer function $f(\boldsymbol{x})$, and $\text{var}(\beta_{\mathbf{0}}) = \tau^2$. For example, in the case of a two-level factor, if we take

$$\boldsymbol{U}_j = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \text{ and } \boldsymbol{\Psi}_j = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \tag{2.5.2}$$

then $\boldsymbol{R}_t = \text{diag}(1, r, \ldots, r, r^2, \ldots, r^2, \ldots, r^p)$, where $\tau^2 = \sigma_0^2 2^{-p}(1+r)^p$, $r = (1-\rho)/(1+\rho)$. This is exactly the same $\boldsymbol{R}_t$ matrix used in Section 2.4.

For factors with more than two levels, we can similarly obtain the prior variance-covariance matrix by appropriately choosing the $\boldsymbol{U}_j$ and $\boldsymbol{\Psi}_j$ matrices. The orthonormal contrast coefficient matrix for a three-level factor is given by

$$\boldsymbol{U}_j = \begin{pmatrix} 1 & -\frac{\sqrt{6}}{2} & \frac{\sqrt{2}}{2} \\ 1 & 0 & -\sqrt{2} \\ 1 & \frac{\sqrt{6}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}. \tag{2.5.3}$$

The choice of correlation matrix should depend on the type of factor, viz. qualitative or quantitative. Joseph and Delaney (2007) suggested using an isotropic correlation function for qualitative factors and a Gaussian correlation function for quantitative factors. Thus,

the correlation matrices for qualitative and quantitative factors are

$$\boldsymbol{\Psi}_j = \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix} \text{ and } \boldsymbol{\Psi}_j = \begin{pmatrix} 1 & \rho & \rho^4 \\ \rho & 1 & \rho \\ \rho^4 & \rho & 1 \end{pmatrix}, \tag{2.5.4}$$

respectively. The functionally induced prior can be shown to satisfy the effect hierarchy principle in assumption (ii). Therefore, now we can specify a prior for the block effects to satisfy assumptions (iii) and (iv). By the same arguments made in Section 2.4, this can be achieved by simply choosing $\boldsymbol{\beta}_b \sim N(\mathbf{0}, \tau^2\mathbf{I})$. Thus, B-optimal design can be obtained by maximizing $B(D)$ in (2.4.1) with $\rho = 1/2$, which corresponds to using $r = 1/3$ in the two-level case.

Example 3. Consider a blocked $(36, 2^2 3^{1+2} : 2^1 3^1)$ design in six blocks for an experiment containing two two-level factors, one three-level qualitative factor, and two three-level quantitative factors. The B-optimal design D_5 is shown in Table 2.5.1, where the two levels of two-level factors are coded by ± 1 and the three levels of the three-level factors are coded by $-1, 0$ and 1. For comparison, an $OA(36, 2^3 3^4)$-based blocked design is also presented in the same table denoted by D_6, in which the first two columns are assigned to the two two-level factors, the third column is the three-level qualitative three-level factor, the next two columns are the two quantitative three-level factors, and the last two columns are the blocking two- and three-level columns. Note that both designs are balanced, but D_5 is not orthogonal. The relative efficiency plot in Figure 2.5.1 shows that D_5 is uniformly better than D_6 for all $\rho \in (0, 1)$.

Similar to the previous two examples, we compute the average absolute correlations of the treatment main effects, two-factor interactions and block effects. Note that here we decompose each three-level factor into two components as in model matrix (2.5.3). Table 2.5.2 shows that the B-optimal design D_5 has less contamination on the treatment effects and slightly larger contamination on the block effects, compared to the orthogonal design D_6. Thus, D_5 can estimate the treatment effects more precisely and therefore, it is a better

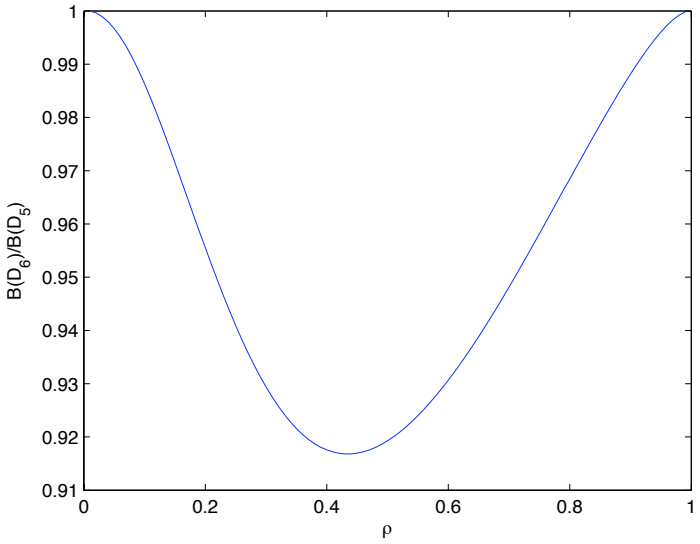


Figure 2.5.1: Relative efficiency: $B(D_6)/B(D_5)$.

blocked design than D_6.

2.6 Concluding Remarks

We proposed a Bayesian optimality criterion for the optimal blocking of two-level designs. The proposed B-criterion helps estimate treatment effects more precisely by sacrificing the precision of block effects. The B-criterion is found to be in close agreement with the four existing criteria for the regular two-level designs. In fact, B-criterion completely agrees with the four criteria for the 8-run designs. It differs with three of the criteria W_{SCF}, W_2, and W_{CC} for two cases in 16-run designs and four cases in 32-run designs. This shows that the proposed B-criterion is closer to the W_1 criterion. When they are in disagreement, it is shown through examples that the B-criterion better satisfy the objectives of blocking.

Although the B-criterion is similar to the existing criteria for the regular designs, it is more general. It can be applied to nonregular designs as well as to designs with any number of runs. This generality is not shared by the existing criteria and is a great advantage of the proposed criterion. The proposed Bayesian approach is also more flexible to handle different types of factors and factors with more than two levels.

Table 2.5.1: B-optimal design D_5 and $OA(36, 2^3 3^4)$-based design D_6 in Example 3

	D_5							D_6						
Run	t_1	t_2	t_3	t_4	t_5	b_1	b_2	t_1	t_2	t_3	t_4	t_5	b_1	b_2
1	−1	1	1	−1	−1	−1	−1	1	−1	−1	−1	0	1	−1
2	1	−1	−1	0	−1	−1	−1	1	1	−1	−1	−1	1	−1
3	1	1	0	0	−1	−1	−1	−1	1	−1	0	−1	−1	−1
4	−1	1	0	1	0	−1	−1	1	−1	−1	1	1	−1	−1
5	−1	−1	0	−1	1	−1	−1	1	1	0	1	1	−1	−1
6	1	−1	1	1	1	−1	−1	1	1	0	1	0	1	−1
7	−1	−1	0	−1	−1	1	−1	−1	1	0	−1	−1	−1	−1
8	1	−1	1	1	−1	1	−1	−1	−1	0	0	1	1	−1
9	−1	1	1	1	−1	1	−1	−1	−1	1	0	1	1	−1
10	1	1	−1	−1	1	1	−1	1	−1	1	0	−1	−1	−1
11	1	−1	1	−1	1	1	−1	−1	1	1	1	0	1	−1
12	−1	−1	0	1	1	1	−1	−1	−1	1	−1	0	−1	−1
13	1	1	−1	−1	−1	−1	0	1	−1	0	0	1	1	0
14	−1	−1	0	0	0	−1	0	1	1	0	0	0	1	0
15	−1	1	1	0	0	−1	0	−1	1	0	1	0	−1	0
16	1	−1	0	1	0	−1	0	1	−1	0	−1	−1	−1	0
17	1	1	1	1	0	−1	0	1	1	1	−1	−1	−1	0
18	−1	1	−1	0	1	−1	0	1	1	1	−1	1	1	0
19	1	−1	1	−1	−1	1	0	−1	1	1	0	0	−1	0
20	−1	1	0	−1	0	1	0	−1	−1	1	1	−1	1	0
21	1	1	−1	0	0	1	0	−1	−1	−1	1	−1	1	0
22	−1	−1	−1	1	0	1	0	1	−1	−1	1	0	−1	0
23	1	−1	−1	0	1	1	0	−1	1	−1	−1	1	1	0
24	1	1	0	0	1	1	0	−1	−1	−1	0	1	−1	0
25	−1	1	−1	0	−1	−1	1	1	−1	1	1	−1	1	1
26	−1	−1	1	0	−1	−1	1	1	1	1	1	1	1	1
27	1	−1	0	−1	0	−1	1	−1	1	1	−1	1	−1	1
28	1	1	1	−1	0	−1	1	1	−1	1	0	0	−1	1
29	−1	−1	1	0	1	−1	1	1	1	−1	0	0	−1	1
30	1	1	−1	1	1	−1	1	1	1	−1	0	−1	1	1
31	1	1	−1	1	−1	1	1	−1	1	−1	1	1	−1	1
32	−1	−1	0	1	−1	1	1	−1	−1	−1	−1	0	1	1
33	−1	−1	−1	−1	0	1	1	−1	−1	0	−1	0	1	1
34	1	−1	1	0	0	1	1	1	−1	0	−1	1	−1	1
35	−1	1	1	−1	1	1	1	−1	1	0	0	−1	1	1
36	−1	1	1	1	1	1	1	−1	−1	0	1	−1	−1	1

Table 2.5.2: Average absolute correlations of the effects of D_5 and D_6

Design	Main effects	Two-factor interactions	Block effects
D_5	0.0505	0.0607	0.1067
D_6	0.0540	0.0894	0.1005

CHAPTER III

A NEW MODELING APPROACH FOR MIXTURE-OF-MIXTURES EXPERIMENTS

3.1 Introduction

In some mixture experiments the mixture components themselves are made up of other sub-components, or more generally, the mixture components can be divided into different categories or groups. These types of mixture experiments have been called *mixture-of-mixtures* experiments (Piepel, 1999) or *categorized-component* mixture experiments (Cornell and Ramsey, 1998). Each mixture component or category is called a *major component* and the mixture components contained in the major components or categories are called *minor components*. The mixture-of-mixtures experiments seem to have arisen more and more frequently in practice. See Piepel (1999), Dingstad et. al. (2003), Borges et. al. (2007), and Didier et. al. (2007) for several case studies in pharmaceutical development, food production, and chemical formulation.

Our research is primarily motivated by a mixture-of-mixtures experiment conducted to formulate a new kind of Pringles® potato crisp, whose package form is changed from a can to a bag. There are three major components A, B, and C. The major component A is composed by two minor components A_1 and A_2, and B is composed by two minor components B_1 and B_2. Component C is pure material, which can be considered to have only a single minor component. There are several characteristics observed as experimental outputs. Among them, we focus on the hardness of the potato crisp (Hardness) and the percentage of fat (% Fat). The main objective is to increase the hardness of the new potato crisp so that the crisp will not easily break in the bag. It is known that, hardness can be increased by increasing the starch content. As a result, the optimization may lead to a formulation containing a larger percentage of fat that may not be acceptable to the consumers. Therefore,

the percentage of fat is also an important characteristic which should be minimized.

Assume that there are M major components and let c_i be the proportion of the ith major component contributed to the whole mixture. They meet the constraint:

$$\sum_{i=1}^{M} c_i = 1, \quad 0 \le c_i \le 1, \quad i = 1, 2, \ldots, M. \tag{3.1.1}$$

The ith major component is composed of m_i ($m_i \ge 1$) minor components. Denote the proportions of these minor components contributed to the ith major component as x_{ij} for $j = 1, \ldots, m_i$. The notation is shown graphically in Figure 3.1.1. The minor component proportions should satisfy the constraints:

$$\sum_{j=1}^{m_i} x_{ij} = 1, \quad \text{and} \quad 0 \le x_{ij} \le 1 \tag{3.1.2}$$
$$i = 1, 2, \ldots, M; \quad j = 1, 2, \ldots, m_i.$$

In general, there could be additional bounds and linear constraints (3.1.3) and (3.1.4) on the major and minor components:

$$L_i \le c_i \le U_i, \quad i = 1, 2, \ldots, M, \tag{3.1.3}$$
$$C_k \le \textstyle\sum_{i=1}^{M} a_{i,k} c_i \le D_k, \quad \text{for the } k\text{th constraint.}$$

$$l_{ij} \le x_{ij} \le u_{ij}, \quad i = 1, 2, \ldots, M; j = 1, 2, \ldots, m_i, \tag{3.1.4}$$
$$c_k \le \textstyle\sum_{i=1}^{M} \sum_{j=1}^{m_i} b_{k,i,j} x_{ij} \le d_k, \quad \text{for the } k\text{th constraint.}$$

For the Pringles® experiment, $M = 3$, $m_1 = m_2 = 2$ and $m_3 = 1$. Its constraints are given by

$$\begin{aligned} c_1 + c_2 + c_3 &= 1, & 0.601 \le c_1 &\le 0.643, \\ 0.34 \le c_2 &\le 0.38, & 0.017 \le c_3 &\le 0.019, \\ x_{11} + x_{12} &= 1, & x_{21} + x_{22} &= 1, \\ 0.835 \le x_{11} &\le 0.905, & 0.9 \le x_{21} &\le 0.98, \\ 0.095 \le x_{12} &\le 0.165, & 0.02 \le x_{22} &\le 0.1. \end{aligned} \tag{3.1.5}$$

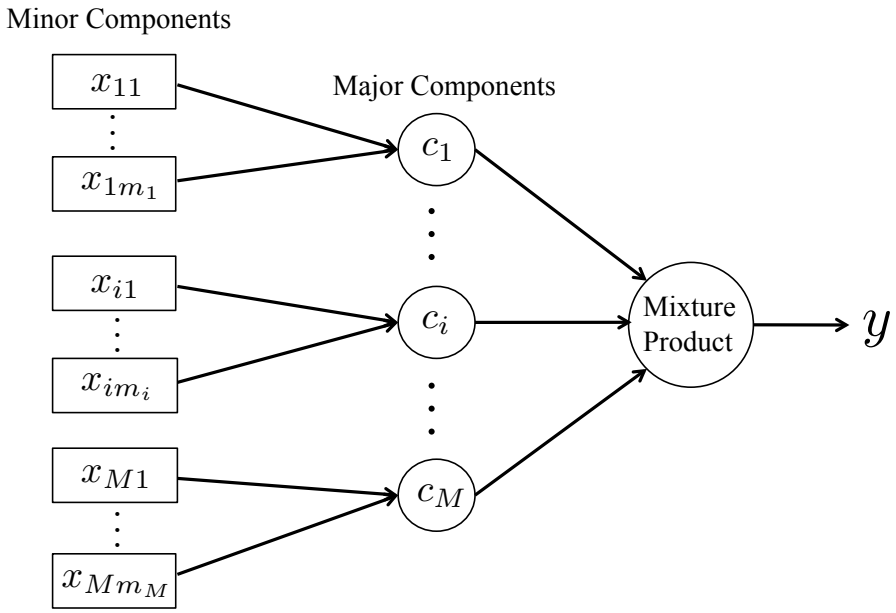


Figure 3.1.1: Mixture-of-Mixtures Structure

The design and analysis of mixture-of-mixtures experiments are more complex than those of the classic mixture experiments, due to the additional constraints (3.1.2) on minor components and the special double-layer-structure of components as shown in Figure 3.1.1. The early works on mixture-of-mixtures experiments assumed that the proportions of the major components are fixed constants. Lambrakis (1968, 1969) first introduced the multiple-Scheffé model and multiple-lattice design strategy. Other methods were also used such as in Cornell and Good (1970) and Cornell (1971). Cornell and Ramsey (1998) extended the multiple-Scheffé model to the case where both the major and minor components can be varied. Since then the multiple-Scheffé model has become very popular. Despite its popularity, the Multiple-Scheffé model has some limitations which will be discussed in the next section.

This paper is organized as follows. In Section 3.2, we first review the multiple-Scheffé model, and then point out some of its limitations for dealing with more general mixture-of-mixtures experiments. Then in Section 3.3, we propose a new modeling approach, which we call the *major-minor model*. We also explain how to interpret the model and compare

it with the multiple-Scheffé model on several aspects. In Section 3.4, we propose a general design strategy for mixture-of-mixtures experiments. In Section 3.5, we apply our proposed design and modeling methods to the Pringles® mixture-of-mixtures experiment. In Section 3.6, we compare the prediction performance of the major-minor model and the multiple-Scheffé model using simulations and the paper concludes with a summary in 3.7.

3.2 Multiple-Scheffé Model

The multiple-Scheffé model was first introduced to study the mixture-of-mixtures experiments in which the proportions of the major components are fixed. Essentially, the multiple-Scheffé model is a product model. Let $f_i(x_{i1}, \ldots, x_{im_i})$ be a mixture model for the minor components of the ith major component. For those major components having only a single component ($m_i = 1$), take $f_i \equiv 1$. When all c_i's are fixed, the multiple-Scheffé model is a product of $f_i(x_{i1}, \ldots, x_{im_i})$:

$$f(\boldsymbol{x}, \boldsymbol{\gamma}) = \prod_{i=1}^{M} f_i(x_{i1}, \ldots, x_{im_i}). \tag{3.2.1}$$

Cornell and Ramsey (1998) generalized the multiple-Scheffé model to the case when the proportions c_i are varied in the experiment. The general multiple-Scheffé model is given by

$$G(\boldsymbol{c}, \boldsymbol{x}, \boldsymbol{\gamma}) = h(c_1, \ldots, c_M) \times \prod_{i=1}^{M} f_i(x_{i1}, \ldots, x_{im_i}). \tag{3.2.2}$$

Here $h(c_1, \ldots, c_M)$ is a mixture model of the major components and $\boldsymbol{\gamma}$ is the vector of unknown parameters in the expanded polynomials. Note that all the models discussed in this paper are linear in the unknown parameters.

In the multiple-Scheffé model, the choices of f_i and h are very flexible. Their forms and orders can be decided freely and independently of each other. The most often used mixture models are the canonical polynomials (without intercepts) called Scheffé model, introduced by Scheffé (1958, 1963). We can also transform the m_i or M mixture components into $m_i - 1$ or $M - 1$ independent variables (as shown in Chapter 3 in Cornell (2003)) so that the Scheffé

models are reparameterized into equivalent regular polynomials with independent variables and intercepts.

Consider the photoresist-coating experiment studied in Cornell and Ramsey (1998). There are two major components R_1 and R_2, each of which has two minor components. In their original paper, the model f_1 and f_2 for minor components are second-order Scheffé models given by:

$$f(x_{i1}, x_{i2}) = \gamma_1 x_{i1} + \gamma_2 x_{i2} + \gamma_3 x_{i1} x_{i2}, \quad \text{for } i = 1, 2. \tag{3.2.3}$$

Thus their product $f(\boldsymbol{x}, \boldsymbol{\gamma}) = \prod_{i=1}^{2} f_i(x_{i1}, x_{i2})$ becomes a nine-term double-Scheffé model:

$$\begin{aligned} f(\boldsymbol{x}, \boldsymbol{\gamma}) = {} & \gamma_1 x_{11} x_{21} + \gamma_2 x_{11} x_{22} + \gamma_3 x_{12} x_{21} + \gamma_4 x_{12} x_{22} + \gamma_5 x_{11} x_{12} x_{21} \\ & + \gamma_6 x_{11} x_{12} x_{22} + \gamma_7 x_{11} x_{21} x_{22} + \gamma_8 x_{12} x_{21} x_{22} + \gamma_9 x_{11} x_{12} x_{21} x_{22}. \end{aligned} \tag{3.2.4}$$

If we use independent variables $z_1 = x_{12} - x_{11}$ and $z_2 = x_{22} - x_{21}$, then (3.2.4) becomes

$$f(\boldsymbol{z}, \boldsymbol{\gamma}') = \gamma'_0 + \gamma'_1 z_1 + \gamma'_2 z_2 + \gamma'_3 z_1 z_2 + \gamma'_4 z_1^2 + \gamma'_5 z_2^2 + \gamma'_6 z_1^2 z_2 + \gamma'_7 z_1 z_2^2 + \gamma'_8 z_1^2 z_2^2. \tag{3.2.5}$$

The model for the major components h is also a second-order Scheffé model. Thus, the multiple-Scheffé model contains $3^3 = 27$ terms given by

$$\begin{aligned} G(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{\gamma}) &= f(\boldsymbol{x}, \boldsymbol{\gamma}_1) c_1 + f(\boldsymbol{x}, \boldsymbol{\gamma}_2) c_2 + f(\boldsymbol{x}, \boldsymbol{\gamma}_3) c_1 c_2, \\ \text{or } G(\boldsymbol{z}, \boldsymbol{c}, \boldsymbol{\gamma}') &= f(\boldsymbol{z}, \boldsymbol{\gamma}'_1) c_1 + f(\boldsymbol{z}, \boldsymbol{\gamma}'_2) c_2 + f(\boldsymbol{z}, \boldsymbol{\gamma}'_3) c_1 c_2. \end{aligned} \tag{3.2.6}$$

Although the multiple-Scheffé model is commonly used in practice, it has some limitations. Multiple-Scheffé model can not be directly applied to the mixture-of-mixtures experiments where the proportion c_i of any major component is allowed to be zero. For instance, in the photoresist-coating experiment, if $(c_1, c_2) = (0, 1)$, then the multiple-Scheffé model (3.2.6) simply becomes (3.2.4) or (3.2.5), which returns the expected response value for pure major component R_2. Note that (3.2.4) and (3.2.5) still contains variables of the minor components of R_1, which is unreasonable because the response cannot depend on the

minor components of R_1 when it is not present in the mixture. A quick fix to this problem is to remove x_{11}, x_{12}, or z_1 from the models when $c_1 = 0$. This approach is equivalent to modifying the model by introducing indicator functions as follows. Let $I(c_i > 0) = 1$ if $c_i > 0$ and 0 otherwise. Then, (3.2.6) becomes

$$\begin{aligned}
&\{\gamma_1 + \gamma_2 z_1 + \gamma_3 I(c_2 > 0) z_2 + \gamma_4 I(c_2 > 0) z_1 z_2 + \gamma_5 z_1^2 + \gamma_6 I(c_2 > 0) z_2^2 \\
&+ \gamma_7 I(c_2 > 0) z_1^2 z_2 + \gamma_8 I(c_2 > 0) z_1 z_2^2 + \gamma_9 I(c_2 > 0) z_1^2 z_2^2\} c_1 + \\
&\{\gamma_{10} + \gamma_{11} I(c_1 > 0) z_1 + \gamma_{12} z_2 + \gamma_{13} I(c_1 > 0) z_1 z_2 + \gamma_{14} I(c_1 > 0) z_1^2 + \\
&\gamma_{15} z_2^2 + \gamma_{16} I(c_1 > 0) z_1^2 z_2 + \gamma_{17} I(c_1 > 0) z_1 z_2^2 + \gamma_{18} I(c_1 > 0) z_1^2 z_2^2\} c_2 + \\
&\{\gamma_{19} + \gamma_{20} z_1 + \gamma_{21} z_2 + \gamma_{22} z_1 z_2 + \gamma_{23} z_1^2 + \gamma_{24} z_2^2 + \gamma_{25} z_1^2 z_2 + \gamma_{26} z_1 z_2^2 + \gamma_{27} z_1^2 z_2^2\} c_1 c_2.
\end{aligned} \tag{3.2.7}$$

This is a discontinuous and nondifferentiable response surface model. It is quite unlikely that such an ill-behaved function will be a physically meaningful representation of the true underlying response surface model.

Another limitation of multiple-Scheffé model can be seen when the major component model h is of first-order. As we have mentioned, according to the product structure of the multiple-Schffé model, the orders of h and f_i's are not related. For the photoresist-coating experiment, if we assume the response is linear with respect to the major components and a first-order Scheffé model $h(\boldsymbol{c}, \boldsymbol{\alpha}) = \alpha_1 c_1 + \alpha_2 c_2$ is used, then (3.2.6) becomes

$$f(\boldsymbol{x}, \boldsymbol{\gamma}) c_1 + f(\boldsymbol{x}, \boldsymbol{\gamma}) c_2,$$

where $f(\boldsymbol{x}, \boldsymbol{\gamma})$ is (3.2.4) or (3.2.5). Because the major component model is first-order, there are no nonlinear blending properties $c_1 c_2$, $c_1 c_3$, and $c_2 c_3$ in the model. However, there are quadratic blending properties between the minors from different majors, such as in term $x_{11} x_{21} c_1$. This is counter-intuitive. Since the minor components are part of major components, if the minor components exhibit nonlinear blending properties, then so should their major components.

Lastly and most importantly, the multiple-Scheffé model can easily become large in size. Its number of terms increases rapidly as the number of minor/major components

increases, or as the models for minor/major components become larger. As a result, prohibitively large experimental designs are required to support the multiple-Scheffé model.

3.3 Major-Minor Model

3.3.1 The General Model Form

The model we propose follows the mixture-of-mixtures structure shown in Figure 3.1.1. It is developed in two stages. First, we use a model $h(c_1, \ldots, c_M)$ to capture the relationship between the mixture characteristic (y) and the major components. Thus, $y = h(c_1, \ldots, c_M) + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. We may choose a Scheffé model of appropriate order for h. The first- and second-order Scheffé models are given by

$$h(c_1, \ldots, c_M) = \sum_{i=1}^{M} \alpha_i c_i,$$

$$h(c_1, \ldots, c_M) = \sum_{i=1}^{M} \alpha_i c_i + \sum_{1 \le i < j \le M} \alpha_{ij} c_i c_j.$$

Because the minor components can be changed to alter the blending properties of the major components, in the second stage we model the coefficients in h as a function of the minor components. As shown in Figure 3.1.1, we assume that the blending property of a major component is affected by only its minor components; the nonlinear blending properties between two major components, if they exist, are affected by only their minor components; and so on. Thus, let $\alpha_i = f_i(z_{i,1}, \ldots, z_{i,m_i-1})$ and $\alpha_{ij} = f_i(z_{i,1}, \ldots, z_{i,m_i-1}) \times f_j(z_{j,1}, \ldots, z_{j,m_j-1})$. Take $f_i \equiv 1$ if $m_i = 1$. Here $z_{i,1}, \ldots, z_{i,m_i-1}$ are the $m_i - 1$ independent variables transformed from $x_{i1}, \ldots, x_{im_i}$. Hence f_i's are regular polynomials with intercepts and independent variables. Thus, the proposed model can be written as

$$G(\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{\gamma}) = \sum_{i=1}^{M} f_i(z_{i,1}, \ldots, z_{i,m_i-1}) c_i \tag{3.3.1}$$

$$G(\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{\gamma}) = \sum_{i=1}^{M} f_i(z_{i,1}, \ldots, z_{i,m_i-1}) c_i + \sum_{1 \le i < j \le M} f_i(z_{i,1}, \ldots, z_{i,m_i-1}) f_j(z_{j,1}, \ldots, z_{j,m_j-1}) c_i c_j \tag{3.3.2}$$

Again, (3.3.1) and (3.3.2) are linear in the coefficients $\boldsymbol{\gamma}$ of the expanded polynomials. If h is chosen to be cubic or higher order Scheffé model, the proposed model can be constructed in the similar way by letting the coefficients of the terms $c_i c_j \dots c_k$ to be the product $f_i(\boldsymbol{z}_i) f_j(\boldsymbol{z}_j) \dots f_k(\boldsymbol{z}_k)$, where $\boldsymbol{z}_i$ is the vector of $z_{i,1}, \dots, z_{i,m_i-1}$. We call this proposed model the *major-minor* model.

A main distinguishing feature of the major-minor model compared to the multiple-Scheffé model is that the models for minor components $f_i(\boldsymbol{z}_i)$'s always appear along with their major component proportions c_i's. Thus, if we need to study the interaction among minor components of different major components, then we must also study the interaction among those major components. This rules out the use of many multiple-Scheffé models including the original multiple-Scheffé model in (3.2.1).

Consider again the photoresist-coating experiment. We can use quadratic polynomials $f_i(z_i, \boldsymbol{\gamma}) = \gamma_0 + \gamma_1 z_i + \gamma_2 z_i^2$ for minor components, and quadratic Scheffé model for major components. Then the major-minor model is given by

$$\begin{aligned} G(\boldsymbol{z}, \boldsymbol{c}, \boldsymbol{\gamma}) = {} & \{\gamma_1 + \gamma_2 z_1 + \gamma_3 z_1^2\} c_1 + \{\gamma_4 + \gamma_5 z_2 + \gamma_6 z_2^2\} c_2 \\ & + \{\gamma_7 + \gamma_8 z_1 + \gamma_9 z_2 + \gamma_{10} z_1 z_2 + \gamma_{11} z_1^2 + \gamma_{12} z_2^2 + \gamma_{13} z_1 z_2^2 + \gamma_{14} z_1^2 z_2 + \gamma_{15} z_1^2 z_2^2\} c_1 c_2. \end{aligned} \tag{3.3.3}$$

Note that when $\boldsymbol{c}$ is fixed, it reduces to the multiple-Scheffeé model in (3.2.5) with the change of notations $\gamma_1 c_1 + \gamma_4 c_2 + \gamma_7 c_1 c_2 \to \gamma_0'$, $\gamma_2 c_1 + \gamma_8 c_1 c_2 \to \gamma_1'$, etc. However, if we were to use the Scheffé model (3.2.3) instead of the regular quadratic polynomial, then the major-minor model becomes

$$\begin{aligned} G(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{\gamma}) = {} & \{\gamma_1 x_{11} + \gamma_2 x_{12} + \gamma_3 x_{11} x_{12}\} c_1 + \{\gamma_4 x_{21} + \gamma_5 x_{21} + \gamma_6 x_{21} x_{22}\} c_2 \\ & + \{\gamma_7 x_{11} x_{21} + \gamma_8 x_{11} x_{22} + \gamma_9 x_{12} x_{21} + \gamma_{10} x_{12} x_{22} + \gamma_{11} x_{11} x_{12} x_{21} \\ & + \gamma_{12} x_{11} x_{12} x_{22} + \gamma_{13} x_{11} x_{21} x_{22} + \gamma_{14} x_{12} x_{21} x_{22} + \gamma_{15} x_{11} x_{12} x_{21} x_{22}\} c_1 c_2. \end{aligned}$$

Now, when $\boldsymbol{c}$ is fixed, it can not be reduced to (3.2.4) by merging the same terms, making the model unidentifiable. Thus, the use of regular polynomials are preferred over Scheffé models for the f_i's.

3.3.2 Comparison With the Multiple-Scheffé Model

The major-minor model is in fact a sub-model of the multiple-Scheffé model. To see this, we just to need to compare the terms containing c_i, c_ic_j, $c_ic_jc_k$, ... of the two models. In the multiple-Scheffé model, these terms are

$$\left\{\prod_{i=1}^{M} f_i(z_i)\right\} c_i, \quad \left\{\prod_{i=1}^{M} f_i(z_i)\right\} c_ic_j, \quad \text{and } \left\{\prod_{i=1}^{M} f_i(z_i)\right\} c_ic_jc_k,$$

which include all their counterparts

$$f_i(z_i)c_i, \quad f_i(z_i)f_j(z_j)c_ic_j, \quad \text{and } f_i(z_i)f_j(z_j)f_k(z_k)c_ic_jc_k$$

of the major-minor model. Thus, the major-minor model omits the terms that are the product of some major component proportions and their non-related minor component proportions, such as $z_{3,1}c_1c_2$, $z_{1,1}z_{2,1}c_1$, etc. A natural question would be why not always consider the multiple-Scheffé model and apply a variable selection technique to obtain a "better" sub-model. However, such a sub-model will be better only in terms of fit to the data; it may not be physically meaningful and may lack predictive power outside the experimental region. Moreover, it is necessary to have a meaningful submodel before obtaining the data for the purpose of designing smaller experiments.

The major-minor model overcomes the limitations of the multiple-Scheffé model mentioned in the previous section. First, if any of the major components is reduced to 0, then all of its minor components are eliminated from the major-minor model. This is a great advantage over the multiple-Scheffé model. In fact, the advantage is not limited to the case of zero major component proportions. Intuition suggests that when the proportion of a particular major component in the mixture is reduced, the effect or influence of its minor components on the final product characteristics should also reduce. This property is satisfied for major-minor model but not for multiple-Scheffé model. Thus, clearly the major-minor model better represents the mixture-of-mixtures structure shown in Figure 3.1.1.

Second, if the major model h is a linear Scheffé model, there would be no nonlinear

Table 3.3.1: Extra number of parameters in multiple-Scheffé model compared to major-minor model.

	$m_1 = 1$	$m_1 = 2$	$m_1 = 3$	$m_1 = 4$
$m_2 = 1$	0	6	15	27
$m_2 = 2$	6	32	71	123
$m_2 = 3$	15	71	155	267
$m_2 = 4$	27	123	267	459

blending terms of the minor components of different major components because they are included only in $f_i(\boldsymbol{z}_i)f_j(\boldsymbol{z}_j)c_ic_j$, $f_i(\boldsymbol{z}_i)f_j(\boldsymbol{z}_j)f_k(\boldsymbol{z}_k)c_ic_jc_k$, etc.

Third, the differences between the sizes of the multiple-Scheffé and the major-minor model increase rapidly as M, m_i, or the orders of the models h and f_i increase. For example, suppose we use quadratic models for f_i and h, then the model sizes of the multiple-Scheffé models and the major-minor models are

$$\left(M + \binom{M}{2}\right)\prod_{i=1}^{M}\left(1 + m_i - 1 + \binom{m_i - 1}{2} + m_i - 1\right),$$

and

$$\sum_{i=1}^{M}\left(1 + m_i - 1 + \binom{m_i - 1}{2} + m_i - 1\right) + \\ + \sum_{i\neq j}^{M}\left(1 + m_i - 1 + \binom{m_i - 1}{2} + m_i - 1\right)\left(1 + m_j - 1 + \binom{m_j - 1}{2} + m_j - 1\right).$$

Table 3.3.1 shows the differences between the two model sizes for different values of m_1 and m_2 for the case with $M = 3$ and $m_3 = 1$. It can be seen that major-minor model's advantage on model size becomes more prominent as the experiments or the models become more complicated. Thus, smaller experimental designs can be used for estimating the major-minor model compared to that of the multiple-Scheffé model.

In the following, we compare the two models using the photoresist-coating experiment. We fit the multiple-Scheffé model in (3.2.6) (denoted as MS) and the major-minor model in (3.3.3) (denoted as MM). The model summaries are shown in Table 3.3.2. R^2_A is the adjusted R^2 value and $R^2_{\mathrm{P}} = 1 - \mathrm{PRESS}/\sum_{i=1}^{n}(y_i - \bar{y})^2$, where PRESS is the sum of squares of the leave-one-out prediction errors. The F value is for checking the adequacy of a reduced

Table 3.3.2: Model comparisons for photoresist-coating experiment ($\hat{\sigma}^2 = 0.1512$).

Model	Model Size	R^2	R^2_A	R^2_{P}	F	p-value
MS	27	0.9994	0.9983	——	——	——
MM	15	0.9961	0.9941	0.9888	6.6944	0.0005
Reduced MM	12	0.9960	0.9945	0.9915	5.5502	0.0010
Reduced MS (1)	15	0.9976	0.9964	0.9935	3.5859	0.0111
Reduced MS (2)	12	0.9967	0.9955	0.9930	4.3771	0.0035

model compared to the complete model and is given by

$$F = \frac{(SSE_r - SSE_c)/(p_c - p_r)}{SSE_c/(n - p_c)},$$

where SSE_c and p_c are the error sum of squares and number of unknown coefficients of the complete models, and SSE_r and p_r are those for the reduced model. Here, we consider the multiple-Scheffé model as the complete model. The p-values of the F-test are shown in the last column.

Since the multiple-Scheffé model is a saturated model (there are only 27 distinct design points), it fits the data perfectly well, as shown by its high R^2 and R^2_A. The R^2_{P} cannot be computed because the multiple-Scheffé model cannot be estimated if we remove any design point from the data.

The major-minor model provides a good fit to the data as well, because its R^2 and R^2_A are only slightly smaller than those of the multiple-Scheffé model. The small p-value=.0005 suggests a statistically significant lack-of-fit for the major-minor model. However, the high R^2_{P} shows that the lack-of-fit errors can be neglected in practice. We performed a backward stepwise regression on the major-minor model by minimizing AIC criterion and PRESS value. In the backward elimination process, the terms in $\sum_{i=1}^{M} \gamma_i c_i$ are not allowed to be removed to ensure the "sum-to-one" constraint of $\boldsymbol{c}$. Application of the two criteria resulted in the same reduced model, which contains 12 terms ($c_1c_2z_1^2z_2$, $c_1c_2z_1z_2^2$, and $c_1c_2z_1$ are removed). We can see that both R^2_A and R^2_P increase when the insignificant terms are removed.

For a fair comparison, we need to reduce the multiple-Scheffé model to the same size

as that of the the major-minor model. We again use backward stepwise regression by removing the term with the lowest partial F value in each step. The reduced models are

$$\begin{aligned}
\text{15-term:}\quad G(\boldsymbol{z},\boldsymbol{c},\boldsymbol{\gamma}) &= \{\gamma_1+\gamma_2 z_1+\gamma_3 z_1^2+\gamma_4 z_1^2 z_2^2\}c_1+\{\gamma_5+\gamma_6 z_1+\gamma_7 z_2+\gamma_8 z_2^2\}c_2 \\
&\quad+\{\gamma_9+\gamma_{10} z_1+\gamma_{11} z_2+\gamma_{12} z_1 z_2+\gamma_{13} z_1^2+\gamma_{14} z_2^2+\gamma_{15} z_1^2 z_2^2\}c_1 c_2, \\
\text{12-term:}\quad G(\boldsymbol{z},\boldsymbol{c},\boldsymbol{\gamma}) &= \{\gamma_1+\gamma_2 z_1+\gamma_3 z_1^2+\gamma_4 z_1^2 z_2^2\}c_1+\{\gamma_5+\gamma_6 z_2+\gamma_7 z_2^2\}c_2 \\
&\quad+\{\gamma_8+\gamma_9 z_2+\gamma_{10} z_1 z_2+\gamma_{11} z_1^2+\gamma_{12} z_2^2\}c_1 c_2.
\end{aligned}$$

As can be seen from Table 3.3.2, the reduced multiple-Scheffé models have only a slight advantage in both fitting and prediction over the major-minor models of the same size.

In summary, multiple-Scheffé models provide better fit to the data, but they contain some terms that are not physically meaningful such as $z_1^2 z_2^2 c_1$ and $z_1 c_2$. As described before, one may introduce indicator functions in the multiple-Scheffé model to remove such terms, but they lead to a nondifferentiable and discontinuous response surface model as shown in Figure 3.3.1(a). On the other hand, the major-minor model is smooth (Figure 3.3.1(b)) and provide almost the same fit as the multiple-Scheffé model. The small differences observed between the two types of models, although statistically significant, need not be practically significant. Thus, the major-minor model could be preferred for the analysis of the photoresist experiment.

3.3.3 Interpretation of the Major-Minor Model

Following the two-stage development of the major-minor model, we use a two-stage approach to model interpretation as well. First, we fix the minor component proportions at the center values of the experimental region and obtain the "marginal" effects of the major component proportions. This provides us with the overall relationship between the mixture product characteristic and the major components. Then, in the second stage, we study the effect of the minor components on their corresponding major components.

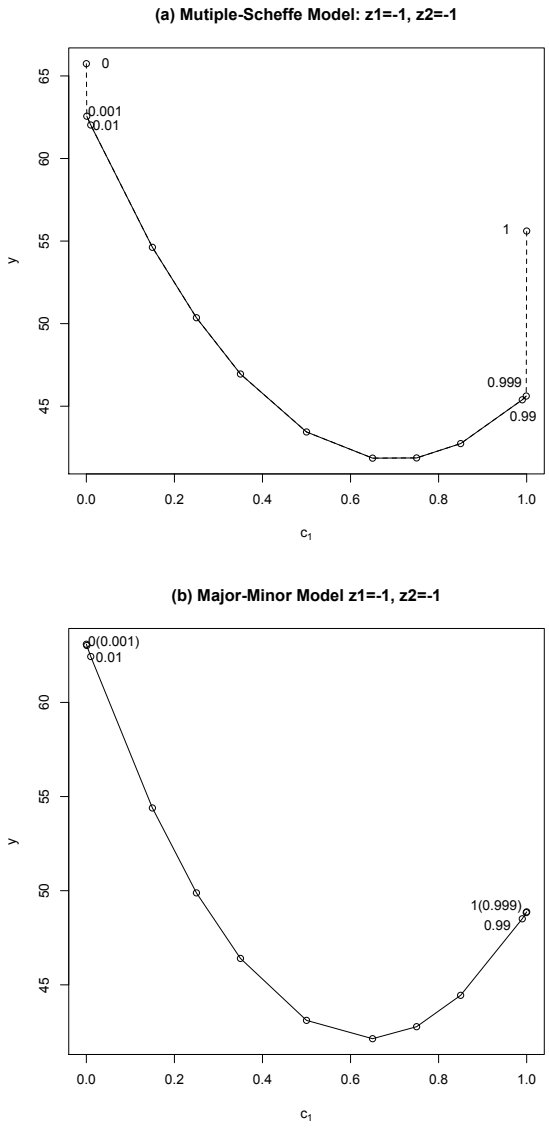


Figure 3.3.1: The response value against c_1 with $z_1 = z_2 = -1$ (a) fitted multiple-Scheffé model (3.2.7) and (b) fitted major-minor model.

Consider the photoresist-coating experiment as an example. The 12-term major-minor model is

$$G(\boldsymbol{z}, \boldsymbol{c}, \boldsymbol{\gamma}) = \{28.32 - 13.72z_1 + 6.84z_1^2\}c_1 + \{35.27 - 20.34z_2 + 7.48z_2^2\}c_2$$
$$+ \{-24.63 + 9.32z_2 + 6.91z_1z_2 - 9.00z_1^2 - 10.52z_2^2 - 5.86z_1^2z_2^2\}c_1c_2.$$

We interpret the terms as follows.

(i) Fix $z_1 = z_2 = 0$. Now, the marginal effects of the major components can be understood from the resulting major component model $28.32c_1 + 35.27c_2 - 24.63c_1c_2$. Thus, the marginal expected response of pure component R_1 is 28.32, the marginal expected response of pure component R_2 is 35.27, and the binary mixture interaction is -24.63.

(ii) The effects of z_1 and z_2 on the expected responses of pure R_1 and R_2 are shown in Figure 3.3.2. We can see that they decrease as z_1 and z_2 increase. The binary mixture interaction of R_1 and R_2 is a quadratic surface of z_1 and z_2 as shown in Figure 3.3.3. The magnitude of the interaction increases as z_1 and z_2 are moved away from the center.

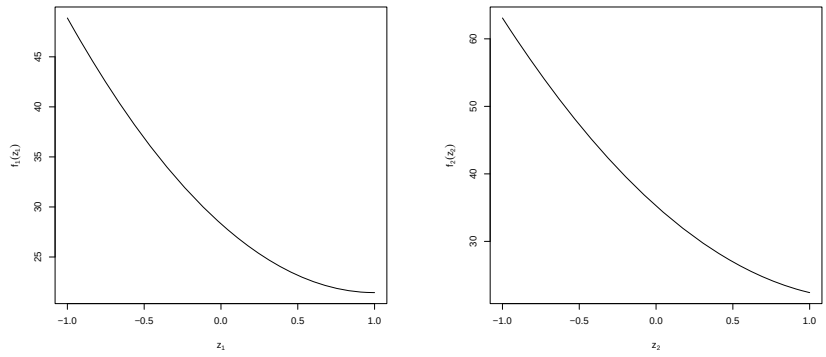


Figure 3.3.2: Expected response value for pure R_1 (left) and R_2 (right).

The interpretation of multiple-Scheffé model in the photoresist experiment can be seen in Cornell and Ramsay (1998). Clearly, the major-minor model is much easier to understand, decipher, and use.

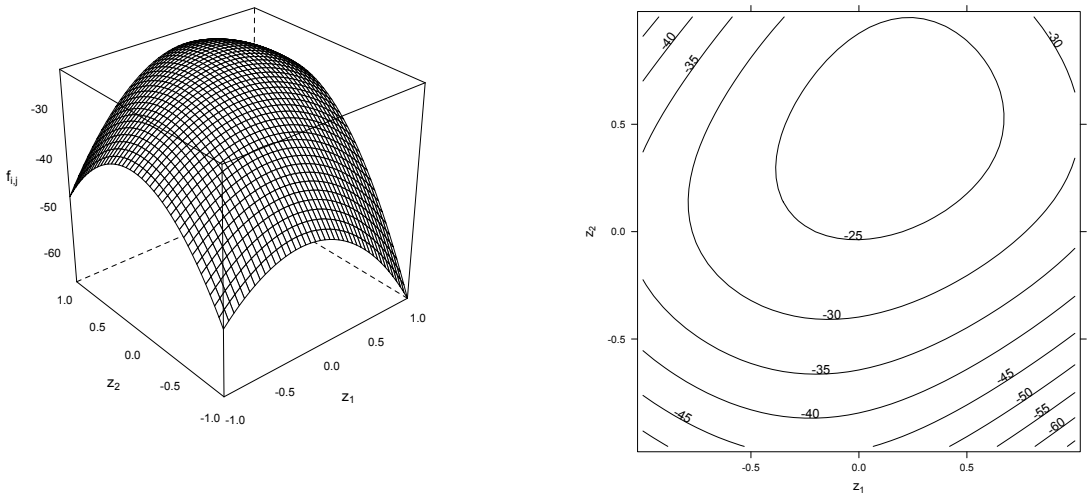


Figure 3.3.3: Binary mixture interaction between R_1 and R_2: surface plot (left) and contour plot (right).

3.4 Experimental Design

Generally, mixture experimental designs are more involved than other types of designs, because the variables of component proportions are constrained and dependent on each other. When the number of components are small and the constraints on the proportions are simple, the design points can be chosen manually using designs such as a simplex design, a simplex-centroid design, and so on. When the number of components are large and the constraints are complicated, computer algorithms are used to generate candidate points from the constrained design space, and design points are chosen from the candidate set according to certain criteria such as *D*- or *A*-optimality.

The following proposition gives an important result for the optimal design of mixture-of-mixtures experiments. The proof is given in the Appendix 3.8.

Proposition 3.4.1. *If D_0, D_1,..., D_M are D-/A-optimal design for models h, f_1, ..., f_M, respectively, then $D = D_0 \otimes D_1 \otimes \cdots \otimes D_M$ is D-/A-optimal design for the multiple-Scheffé model $h \times \prod_{i=1}^{M} f_i$.*

Here $\otimes$ stands for Kronecker product and therefore, $D = D_0 \otimes D_1 \otimes \cdots \otimes D_M$ is a crossed design of D_0, D_1, ..., D_M. Thus, the crossed design is optimal for estimating multiple-Scheffé model. This result is very useful because it is easy to find optimal designs for the small models h, f_1, ..., f_M, which can then be crossed to obtain the optimal design for the large multiple-Scheffé model. The optimal design, otherwise, would have been extremely difficult to find because of its large run size, particularly in a more complicated constrained design space.

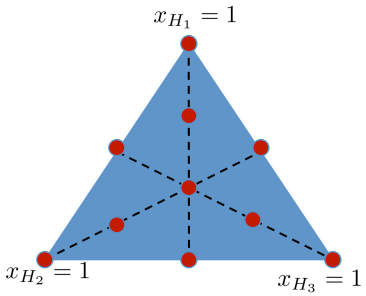
The major-minor model needs only a subset of $D = D_0 \otimes D_1 \otimes \cdots \otimes D_M$ for estimation because it is a sub-model of the multiple-Scheffé model. We can take D as the candidate set and choose design points according to certain optimality criterion. At last, some points for checking the lack-of-fit of the model and some replications for estimating the pure error can be added, if they are affordable.

It is important to point out that although the major-minor model is a sub-model of the multiple-Scheffé model, the design needed for fitting a major-minor model can be quite different from that of a multiple-Scheffé model. For example, the original experiments of Lambrakis (1968, 1969), where major components are fixed and only the minor components are varied are not acceptable as designs for the major-minor model. This is because the interaction between the minor components of two different major components are entertained in the major-minor model only through the interaction term of those two major components and thus, the major components should also be varied in the experiment in order to estimate such interactions.

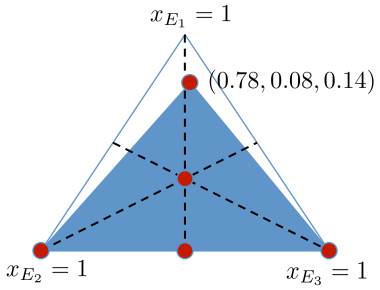
Example 1. Didier et. al. (2007) carried out a mixture-of-mixtures experiment to develop complex culture media used in recombinant protein production. Two categories of components (major components) are involved: hexose (H) and energy provider (E). Hexose is a mixture of three hexoses H_1, H_2, and H_3. Energy provider is a mixture of three energy providers E_1, E_2, and E_3.

In their original experiment, the amounts of the two major components are fixed at

(a) Design for two major components D_0 .



(b) Design for major component H D_1 .



(c) Design for two major component E D_2 .

Figure 3.4.1: *Designs for major and minor components.*

$c_H = 0.831$ and $c_E = 0.169$. Their experimental design is a crossed design of D_1 and D_2 (see Figure 3.4.1). The design D_1 for the hexoses is a 10-point augmented simplex-lattice design, whereas the design D_2 for the energy providers contains three vertices of the constrained triangle, the centroid of the unconstrained triangle, and the middle point of one edge. Thus, their crossed design has 50 runs.

For illustration purposes, we assume that c_H and c_E can be varied at two values $(c_H, c_E) = (0.731, 0.269)$, and $(0.831, 0.169)$ as shown in Figure 3.4.1(a). Denote this design as D_0, which is adequate for estimating a linear Scheffé model h. Following Didier et al. (2007), we assume that f_1 for hexoses to be quadratic and f_2 for energy provider to be linear. Then, the multiple–Scheffé model contains 36 terms, which can be estimated by a design having at least 36 runs. If 100 runs are affordable, then we can use the crossed design $D_0 \otimes D_1 \otimes D_2$. If this is not affordable, we can find a smaller design as follows. It is easy to show that D_0, $\tilde{D}_1$, $\tilde{D}_2$ (shown in Figure 3.4.2) are the D-optimal designs for h, f_1, and f_2. Then by Proposition 1, $D_0 \otimes \tilde{D}_1 \otimes \tilde{D}_2$ is the D-optimal design for the multiple-Scheffé model, which has only 36 runs. To allow for a lack-of-fit test, we can add several replications at the point: $(c_H, c_E, x_{H_1}, x_{H_2}, x_{H_3}, x_{E_1}, x_{E_2}, x_{E_3}) = (0.736, 0.264, 0.33, 0.33, 0.34, 0.33, 0.33, 0.34)$, which is very close to the overall centroid of the constrained design space.

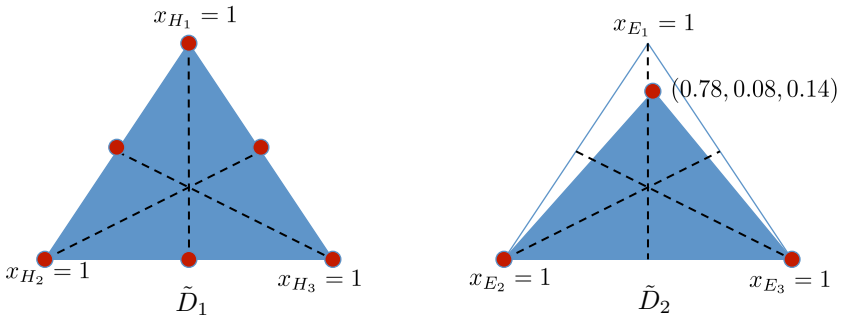


Figure 3.4.2: $\tilde{D}_1$ and $\tilde{D}_2$ in the D-optimal design.

Now consider the major-minor model. It is given by

$$G(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{\gamma}) = (\gamma_1 + \gamma_2 z_{11} + \gamma_3 z_{12} + \gamma_4 z_{11}^2 + \gamma_5 z_{12}^2 + \gamma_6 z_{11} z_{12})c_1 + (\gamma_7 + \gamma_8 z_{21} + \gamma_9 z_{22})c_2,$$

where $z_{i1} = 2x_{i1} - x_{i2} - x_{i3}$ and $z_{i2} = x_{i2} - x_{i3}$ for $i = 1, 2$. The model contains 9 terms. If budget permits, we could use the 36-run crossed design. If not, we can choose a design of size as small as 9 runs. Suppose, we have the budget for 12 runs. Then, we can select the 12 runs from the crossed design $D_0 \otimes D_1 \otimes D_2$ using the D-criterion. The optimal design is shown in Figure 3.4.3 (We used the AlgDesign package in R software.) We can see that the levels are well balanced in the design. Similar to the design for multiple-Scheffé model, we can also add a centroid and some replications to check for the lack-of-fit. This example clearly shows a great advantage of using the major-minor model. We can choose an experimental design of much smaller size than would have required by the multiple–Scheffé model and thus, save time and money for the experimenter.

3.5 Pringles Mixture-of-Mixtures Experiment

Consider the Pringles® mixture-of-mixtures experiment described in the introduction. The objective of the experiment is to create empirical models that can be used for formulation understanding and optimization. Note that this is a preliminary study that was constrained by both time and resources which led to a cap on the number of trails to be set at 17. The constrained design space of the major components is shown as the white area in Figure 3.5.1 and the minor component space is shown in Figure 3.5.2.

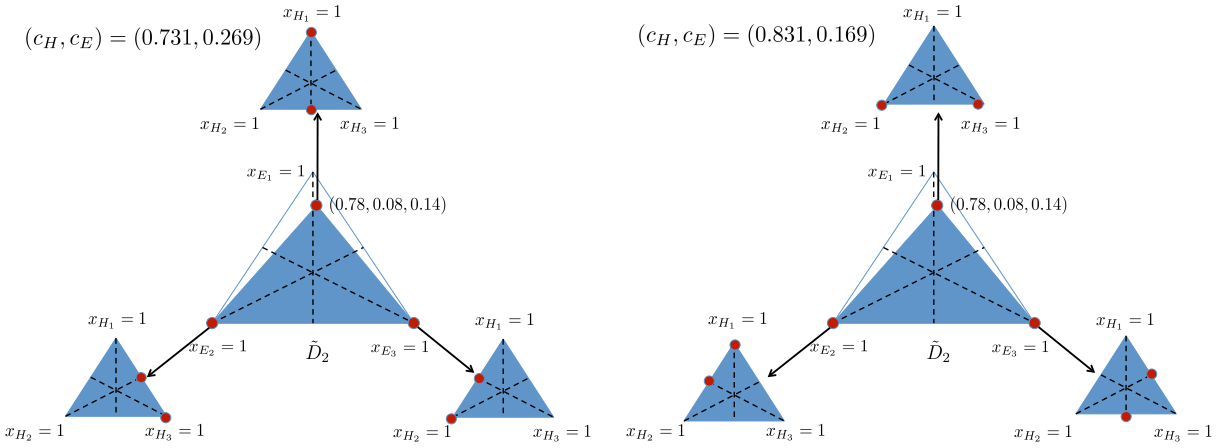


Figure 3.4.3: Design for major-minor model.

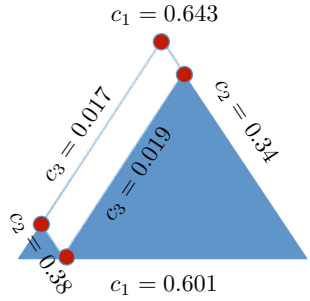


Figure 3.5.1: Major component design space of pringle experiment.

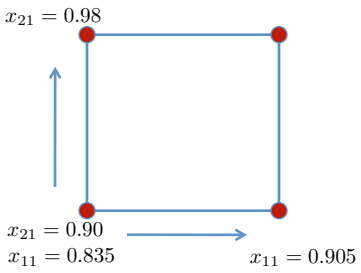


Figure 3.5.2: Minor component design space of pringle experiment.

The multiple-Scheffé model is chosen in which the major component model h along with the minor component models f_1 and f_2 are taken to be linear. The reasons for this choice go hand in hand with the limitations around the time and resources for the study as well as the size of the resulting multiple-Scheffé models. Even with just linear models for all of the components, the multiple-Scheffé model consists of $3 \times 2 \times 2 = 12$ terms. In order to entertain a quadratic model for h and linear models for both f_1 and f_2, the resulting multiple-Scheffé model would consist of $6 \times 2 \times 2 = 24$ terms. Alternatively, a linear model for h and quadratic for either f_1 or f_2 or both would lead to models of size 18 or 27. Each of these scenarios leads to models that have more terms than the required minimum number of 17 trials. This in fact was the initial motivating problem that led to this research for finding alternative models to the multiple-Scheffé model since when the number of major and minor components increases, so does the size and complexity of the multiple-Scheffé model.

The design was chosen so that the design space was sufficiently covered and the model coefficients can be estimated. In order to fit the linear model for the major components, a minimum of three design points in Figure 3.5.1 are sufficient, and these three would clearly be chosen from the four vertices. Therefore, the four-vertice design was chosen so that full coverage of the design space could be obtained. For the minor component design, it is clear that the four vertices shown in Figure 3.5.2 will allow for the estimation of f_1 and f_2. The final design that will support the estimation of the multiple-Scheffé model is simply the crossed design obtained by crossing the four vertices of the major component design with the four vertices of the minor component design. We chose the overall centroid to be the final point as a way to check for curvature. The final design is shown graphically in Figure 3.5.3 and given, along with the resulting data, in Table 3.5.1.

While the multiple-Scheffé model was used with success in the actual experiment, both in the design phase and subsequent analysis phase, this example provides a real data set to compare the model fit for both the multiple-Scheffé model and the proposed major-minor

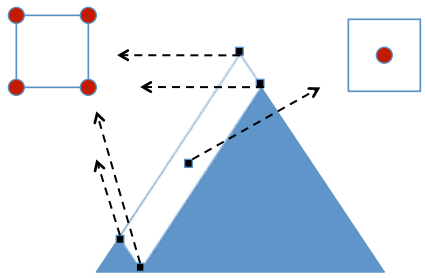


Figure 3.5.3: Experimental design of the pringles experiment.

Table 3.5.1: Pringles mixture-of-mixtures experiment data.

Run	x_{11}	x_{12}	z_1	x_{21}	x_{22}	z_2	c_1	c_2	c_3	% Fat	Hardness
1	0.835	0.165	−1	0.90	0.10	−1	0.603	0.38	0.017	35.040	4.835
2	0.835	0.165	−1	0.90	0.10	−1	0.643	0.34	0.017	32.100	6.375
3	0.835	0.165	−1	0.98	0.02	1	0.603	0.38	0.017	37.800	3.625
4	0.835	0.165	−1	0.98	0.02	1	0.643	0.34	0.017	33.300	5.500
5	0.905	0.095	1	0.90	0.10	−1	0.643	0.34	0.017	31.320	6.875
6	0.905	0.095	1	0.90	0.10	−1	0.603	0.38	0.017	34.026	5.250
7	0.835	0.165	−1	0.90	0.10	−1	0.601	0.38	0.019	34.140	5.000
8	0.835	0.165	−1	0.90	0.10	−1	0.641	0.34	0.019	31.968	6.250
9	0.835	0.165	−1	0.98	0.02	1	0.601	0.38	0.019	36.990	3.625
10	0.905	0.095	1	0.98	0.02	1	0.603	0.38	0.017	35.970	4.250
11	0.905	0.095	1	0.90	0.10	−1	0.601	0.38	0.019	33.870	5.250
12	0.835	0.165	−1	0.98	0.02	1	0.641	0.34	0.019	33.438	4.875
13	0.905	0.095	1	0.98	0.02	1	0.643	0.34	0.017	33.144	4.940
14	0.905	0.095	1	0.90	0.10	−1	0.641	0.34	0.019	32.106	6.165
15	0.905	0.095	1	0.98	0.02	1	0.641	0.34	0.019	33.660	5.565
16	0.905	0.095	1	0.98	0.02	1	0.601	0.38	0.019	35.520	4.875
17	0.87	0.13	0	0.94	0.06	0	0.622	0.36	0.018	33.438	4.875

model. Thus, we consider the following models:

$$\text{MS:}\quad G(\boldsymbol{z},\boldsymbol{c},\boldsymbol{\gamma}) = (\gamma_1+\gamma_2 z_1+\gamma_3 z_2+\gamma_4 z_1 z_2)c_1+(\gamma_5+\gamma_6 z_1+\gamma_7 z_2+\gamma_8 z_1 z_2)c_2 \quad (3.5.1)$$
$$+(\gamma_9+\gamma_{10} z_1+\gamma_{11} z_2+\gamma_{12} z_1 z_2)c_3,$$

$$\text{MM:}\quad G(\boldsymbol{z},\boldsymbol{c},\boldsymbol{\gamma}) = (\gamma_1+\gamma_2 z_1)c_1+(\gamma_3+\gamma_4 z_2)c_2+\gamma_5 c_3, \quad (3.5.2)$$

where $z_1 = (x_{11} - x_{12} - 0.74)/0.07$ and $z_2 = (x_{21} - x_{22} - 0.88)/0.08$. The estimates of the coefficients are shown in Table 3.5.2 and the model summary statistics in Table 3.5.3. For a fair comparison, a 5-term reduced multiple-Scheffé model (denoted as "Reduced MS") is obtained by using backward stepwise regression on the full multiple-Scheffé model. The F-test is the test statistic for comparisons between the sub-models and the multiple-Scheffé model.

From Table 3.5.3, we can see that multiple-Scheffé model gives a good fit to the data. However, it seems to be slightly over fitting because of the noticeable difference between R^2 and R^2_A. The major-minor model also performs well, since its R^2, R^2_A are close to those of the reduced multiple-Scheffé model. The F-test statistics are all smaller than $F_{7,5,0.95}$ indicating no significant difference from the complete model. In terms of R^2_{P}, the major-minor model is comparable to the reduced multiple-Scheffé model and much better than the full model.

Thus, in terms of model fitting, one may consider choosing between the major-minor model and the reduced multiple-Scheffé model. However, in terms of interpretation, the major-minor model is superior. Consider, for example, the models for Hardness. The reduced multiple-Scheffé model contains the terms z_2c_1 and z_1c_2 instead of the terms z_1c_1 and z_2c_2 terms in the major-minor model. Clearly, the terms in the major-minor model are physically more meaningful. Thus, we should choose the major-minor model for optimizing the mixture.

The optimization can be performed using any standard optimization software. The

Table 3.5.2: Coefficients estimations.

	Response: %Fat			Response: Hardness		
Term	MS	Reduced MS	MM	MS	Reduced MS	MM
c_1	10.014	10.014	10.014	16.299	16.299	16.299
c_2	79.764	79.764	79.764	−14.435	−14.435	−14.435
c_3	−52.986	−52.986	−52.986	13.487	13.487	13.487
z_1c_1	1.487		−0.505	−3.317		0.303
z_2c_1	−2.368			−2.932	−0.881	
z_1c_2	−11.039	−0.932		2.917	0.553	
z_2c_2	7.495	2.670	2.670	−0.353		−1.506
z_1c_3	151.487			66.995		
z_2c_3	−15.118			78.006		
$z_1z_2c_1$	4.022			−4.577		
$z_1z_2c_2$	−4.490			0.095		
$z_1z_2c_3$	−53.728			159.485		

Table 3.5.3: Model comparisons for the Pringles experiment.

	Response: %Fat			Response: Hardness		
	MS	Reduced MS	MM	MS	Reduced MS	MM
Model Size	12	5	5	12	5	5
MSE	0.2302	0.2777	0.2963	0.0692	0.1148	0.1280
R^2	0.9775	0.9349	0.9305	0.9731	0.8929	0.8806
R^2_A	0.9280	0.9132	0.9074	0.9138	0.8573	0.8409
F	———	1.3538	1.4918	———	2.1273	2.4536
R^2_{P}	0.7362	0.8689	0.8602	0.6888	0.7834	0.7576

optimal setting to maximize Hardness is given by

$$(x_{11}, x_{12}, x_{21}, x_{22}, c_1, c_2, c_3) = (0.905, 0.095, 0.9, 0.1, 0.643, 0.34, 0.017)$$

and the optimal setting to to minimize %Fat is given by

$$(x_{11}, x_{12}, x_{21}, x_{22}, c_1, c_2, c_3) = (0.905, 0.095, 0.9, 0.1, 0.641, 0.34, 0.019).$$

Incidently, the optimal settings turned out to be run # 5 and run # 14. Because maximizing the hardness of the crisp is of greater importance than reducing the percentage of fat, we simply choose the settings of run # 5 as the new formulation, which maximizes the hardness as well as gives an acceptable percentage of fat. As mentioned before, this experiment is a preliminary study used to determine some directions to go with a more comprehensive follow-up study. For the follow-up study, we should use quadratic models and conduct larger experiments around the current optimal settings to find better formulations.

3.6 Simulation Study

In this section, we study the prediction ability of the major-minor model compared to the multiple-Scheffé model using simulations. Consider a mixture-of-mixtures experiment that has two major components A and B, each of which has two minor components A_1, A_2 and B_1, B_2, whose corresponding proportions are X_{A_1}, X_{A_2}, X_{B_1}, and X_{B_2}. They satisfy the constraint $X_{A_1} + X_{A_2} + X_{B_1} + X_{B_2} = 1$. The following full quadratic and cubic Scheffé mixture models are used as test functions:

I. $f_I(\boldsymbol{X}) = 2X_{A_1} + X_{A_2} + 3X_{B_1} + X_{B_2} + 4(X_{A_1}X_{A_2} + X_{A_1}X_{B_1} + X_{A_1}X_{B_2} + X_{A_2}X_{B_1} + X_{A_2}X_{B_2} + X_{B_1}X_{B_2})$, and

II. $f_{II}(\boldsymbol{X}) = X_{A_1} + X_{A_2} + X_{B_1} + X_{B_2} + 5(X_{A_1}X_{A_2} + X_{A_1}X_{B_1} + X_{A_1}X_{B_2} + X_{A_2}X_{B_1} + X_{A_2}X_{B_2} + X_{B_1}X_{B_2}) + 10(X_{A_1}X_{A_2}X_{B_1} + X_{A_1}X_{A_2}X_{B_2} + X_{A_1}X_{B_1}X_{B_2} + X_{A_2}X_{B_1}X_{B_2})$.

These models are neither multiple-Scheffé models nor major-minor models and thus, a fair comparison can be made using simulations.

Table 3.6.1: Design for major components for simulation study.

Run	c_1	c_2
1	0.25	0.75
2	0.375	0.625
3	0.5	0.5
4	0.625	0.375
5	0.75	0.25

Table 3.6.2: Design for minor components for simulation study.

Run	x_{11}	x_{12}	x_{21}	x_{22}	z_1	z_2
1	0	1	0	1	-1	-1
2	0	1	0.5	0.5	-1	0
3	0	1	1	0	-1	1
4	0.5	0.5	0	1	0	-1
5	0.5	0.5	0.5	0.5	0	0
6	0.5	0.5	1	0	0	1
7	1	0	0	1	1	-1
8	1	0	0.5	0.5	1	0
9	1	0	1	0	1	1

We use quadratic f_1, f_2, and h for both multiple-Scheffé and major-minor models. The resulting models are given in (3.2.6) and (3.3.3). The designs for major and minor components are shown in Tables 3.6.1 and 3.6.2, respectively. The final design is the crossed design of these two designs. We conduct 100 simulations. In each simulation, we simulate data using $y = f_k(\boldsymbol{X}) + \epsilon$, $k = I, II$ from the design, where $X_{A_1} = x_{11}c_1$, $X_{A_2} = x_{12}c_1$, $X_{B_1} = x_{21}c_2$, $X_{B_2} = x_{22}c_2$, and $\epsilon \sim N(0, 1)$. Then, we fit the model (3.2.6) and (3.3.3). To make a fair comparison, we reduce the multiple-Scheffé model to the same size of the major-minor model using backward stepwise regression. Then the reduced multiple-Scheffé and major-minor models are used to calculate the mean squared prediction errors $MSPE = 1/n \sum_{j=1}^{n} (f_k(\boldsymbol{X}_j) - \hat{y}_j)^2$, where the $\boldsymbol{X}_j$'s are sampled from two regions: $c_i \in [0.25, 0.75]$ and $c_i \in [0, 1]$, for $i = 1, 2$. Figure 4.4.3 shows the density plots of the $MSPE$'s of the two models. The left column shows $MSPE$'s when c_i is in $[0.25, 0.75]$ and right for $[0, 1]$.

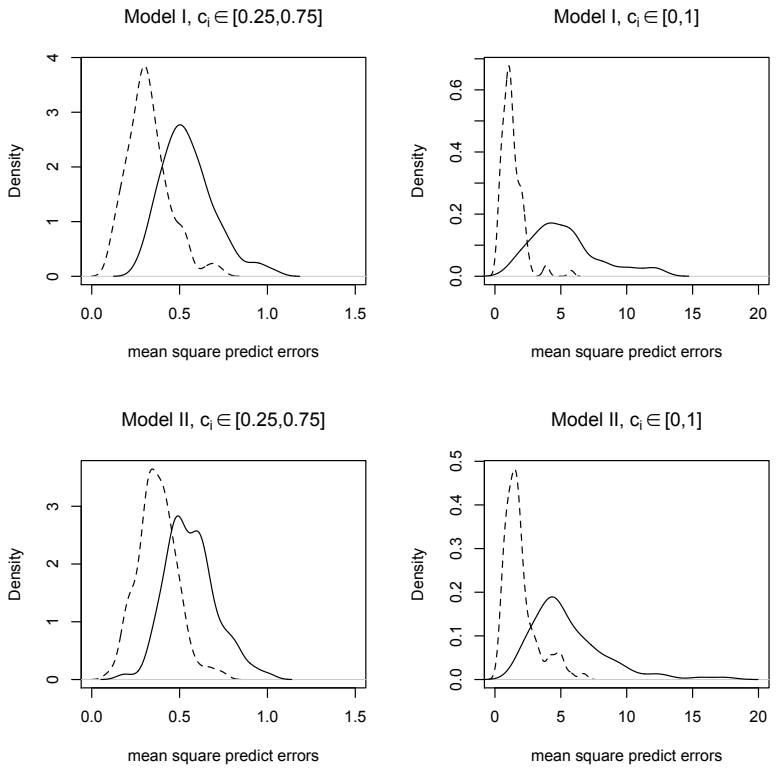


Figure 3.6.1: Comparison of multiple-Scheffé model (solid) and major-minor model (dashed) for the 100 simulated data sets from Models I and II with two boundary conditions.

It can be seen that the major-minor model outperforms the multiple-Scheffé model in all cases. The improvement is much higher when the model is extrapolated outside the experimental region $[0.25, 0.75]$. It further confirms that the performance of multiple-Scheffé model deteriorates as the major components take values closer to 0. This is a key advantage for those doing experiments with several major components with lower constraints equal to 0.

3.7 Conclusions

The multiple-Scheffé model has certain limitations for use in a general mixture-of-mixtures problem. It cannot be used when some of the major component proportions are zero, its interpretations are problematic, and it can become large in size leading to prohibitively large experimental designs. In this article, we have introduced a new type of model called

the major-minor model to overcome these limitations. The new model is smaller and thus, can be estimated with less experimental effort. Moreover, the new model better captures the mixture-of-mixtures structure and can work even when some of the major components are absent in the mixture. We have also proposed a D/A-optimal-based strategy to efficiently design mixture-of-mixtures experiments. The advantages of the major-minor model are demonstrated using two real experiments and simulations.

The major-minor model is shown to be a sub-model of the multiple-Scheffé model. However, this sub-model is obtained by removing the terms that do not make any physical sense. Therefore, although a data-driven model reduction technique can be applied to the multiple-Scheffé model to obtain a sub-model, it need not always result in a major-minor model. Moreover, it is essential to work with a meaningful submodel before obtaining the data so that we can design a smaller experiment. Analysis of several examples show that very little is lost in terms of fit to the data, confirming the validity of the proposed submodel. In summary, we recommend the major-minor model over the multiple-Scheffé model for use in practice.

3.8 Appendix: Proof of Proposition

Without loss of generality, we assume the product model $\boldsymbol{f}(\boldsymbol{x}) : \mathscr{X} \to \mathbb{R}^p$ is a continuous function. It can be written as:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}^1(\boldsymbol{x}^1) \otimes \boldsymbol{f}^2(\boldsymbol{x}^2) \otimes \cdots \otimes \boldsymbol{f}^k(\boldsymbol{x}^k) = \bigotimes_{i=1}^{k} \boldsymbol{f}^i(\boldsymbol{x}^i),$$

where $\otimes$ stands for Kronecker product, $\boldsymbol{f}^i(\boldsymbol{x}^i) : \mathscr{X}_i \to \mathbb{R}^{p_i}$ is a continuous function defined on $\mathscr{X}_i$, and $\prod_{i=1}^{k} p_i = p$. Here $\mathscr{X} \subset \mathbb{R}^d$ is a compact design space, which is a Cartesian product of k compact sets $\mathscr{X}_i \subset \mathbb{R}^{d_i}$, for $i = 1, \ldots, k$, i.e., $\mathscr{X} = \mathscr{X}_1 \times \mathscr{X}_2 \times \cdots \times \mathscr{X}_k$ and $\sum_{i=1}^{k} d_i = d$. In the proof we use continuous design, which is represented by measure ξ over $\mathscr{X}$. The information matrix is defined as:

$$\boldsymbol{M}(\xi) = \int_{\mathscr{X}} \boldsymbol{f}^T(\boldsymbol{x}) \boldsymbol{f}(\boldsymbol{x}) \xi(d\boldsymbol{x}).$$

Let ξ_i be the design for set $\mathscr{X}_i$ and $\boldsymbol{M}(\xi_i)$ be the information matrix of model $\boldsymbol{f}^i$. If ξ is a product measure of $\xi_1, \xi_2, \ldots, \xi_k$, then it is also a crossed design for the design space $\mathscr{X}$, and the information matrix can be written as

$$\begin{aligned}
\boldsymbol{M}(\xi) &= \int_{\mathscr{X}} \boldsymbol{f}^T(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})\xi(d\boldsymbol{x}) \\
&= \int_{\mathscr{X}} (\bigotimes_{i=1}^{k} \boldsymbol{f}^i(\boldsymbol{x}^i))^T (\bigotimes_{i=1}^{k} \boldsymbol{f}^i(\boldsymbol{x}^i)) \prod_{i=1}^{k} \xi_i(d\boldsymbol{x}^i) \\
&= \int_{\mathscr{X}} \bigotimes_{i=1}^{k} (\boldsymbol{f}^i(\boldsymbol{x}^i))^T \boldsymbol{f}^i(\boldsymbol{x}^i) \prod_{i=1}^{k} \xi_i(d\boldsymbol{x}^i) \\
&= \bigotimes_{i=1}^{k} \int_{\mathscr{X}_i} (\boldsymbol{f}^i(\boldsymbol{x}^i))^T \boldsymbol{f}^i(\boldsymbol{x}^i) \xi_i(d\boldsymbol{x}^i) \\
&= \boldsymbol{M}_1(\xi_1) \otimes \cdots \otimes \boldsymbol{M}_k(\xi_k).
\end{aligned}$$

Let ξ_i^* be the D-optimal design of model $\boldsymbol{f}^i$ on $\mathscr{X}_i$. Then, by the general equivalence theorem (see Atkinson and Donev 1992)

$$\min_{\mathscr{X}_i} \boldsymbol{f}^i(\boldsymbol{x}^i)\boldsymbol{M}_i^{-1}(\xi_i^*)(\boldsymbol{f}^i(\boldsymbol{x}^i))^T = p_i.$$

Now, if the design ξ^* is a product measure of $\xi_1^*, \xi_2^*, \ldots, \xi_k^*$, then

$$\begin{aligned}
&\min_{\mathscr{X}} \boldsymbol{f}(\boldsymbol{x})\boldsymbol{M}^{-1}(\xi^*)\boldsymbol{f}^T(\boldsymbol{x}) \\
=&\min_{\mathscr{X}} \{\bigotimes_{i=1}^{k} \boldsymbol{f}^i(\boldsymbol{x}^i)\}\{\bigotimes_{i=1}^{k} \boldsymbol{M}_i^{-1}(\xi_i^*)\}\{\bigotimes_{i=1}^{k} \boldsymbol{f}^i(\boldsymbol{x}^i)\}^T \\
=&\min_{\mathscr{X}} \bigotimes_{i=1}^{k} \{\boldsymbol{f}^i(\boldsymbol{x}^i)\boldsymbol{M}_i^{-1}(\xi_i^*)(\boldsymbol{f}^i(\boldsymbol{x}^i))^T\} \\
=&\prod_{i=1}^{k} \min_{\mathscr{X}_i} \{\boldsymbol{f}^i(\boldsymbol{x}^i)\boldsymbol{M}_i^{-1}(\xi_i^*)(\boldsymbol{f}^i(\boldsymbol{x}^i))^T\} \\
=&p_1 \ldots p_k = p.
\end{aligned}$$

Thus, the crossed design ξ^* is D-optimal design for model $\boldsymbol{f}$.

Similarly, if ξ_i^* is the A-optimal design of model $\boldsymbol{f}^i$ on $\mathscr{X}_i$, then by the general equivalence theorem

$$\min_{\mathscr{X}_i} \boldsymbol{f}^i(\boldsymbol{x}^i)\boldsymbol{M}_i^{-1}(\xi^*)\boldsymbol{M}_i^{-1}(\xi_i^*)(\boldsymbol{f}^i(\boldsymbol{x}^i))^T = tr(\boldsymbol{M}_i^{-1}(\xi_i^*)).$$

Then,

$$
\begin{aligned}
&\min_{\mathscr{X}} \boldsymbol{f}(\boldsymbol{x})\boldsymbol{M}^{-1}(\xi^*)\boldsymbol{M}^{-1}(\xi^*)\boldsymbol{f}^T(\boldsymbol{x}) \\
&= \min_{\mathscr{X}}\{\bigotimes_{i=1}^{k} \boldsymbol{f}^i(\boldsymbol{x}^i)\}\{\bigotimes_{i=1}^{k} \boldsymbol{M}_i^{-1}(\xi_i^*)\}\{\bigotimes_{i=1}^{k} \boldsymbol{M}_i^{-1}(\xi_i^*)\}\{\bigotimes_{i=1}^{k} \boldsymbol{f}^i(\boldsymbol{x}^i)\}^T \\
&= \min_{\mathscr{X}} \bigotimes_{i=1}^{k}\{\boldsymbol{f}^i(\boldsymbol{x}^i)\boldsymbol{M}_i^{-1}\boldsymbol{M}_i^{-1}(\xi_i^*)(\boldsymbol{f}^i(\boldsymbol{x}^i))^T\} \\
&= \prod_{i=1}^{k} tr(\boldsymbol{M}_i^{-1}(\xi_i^*)) = tr(\bigotimes_{i=1}^{k} \boldsymbol{M}_i^{-1}(\xi_i^*)) = tr(\boldsymbol{M}^{-1}(\xi^*)).
\end{aligned}
$$

Thus, the crossed design ξ^* is A-optimal design for model $\boldsymbol{f}$.

CHAPTER IV

REGRESSION-BASED INVERSE DISTANCE WEIGHTING WITH APPLICATIONS TO COMPUTER EXPERIMENTS

4.1 Introduction

We consider the problem of finding an interpolating function through the data $(\boldsymbol{x}_1, y_1)$, ..., $(\boldsymbol{x}_n, y_n)$, where $\boldsymbol{x}_i \in \mathbb{R}^p$. Multivariate interpolation has immense applications in science and engineering; here we focus on the modeling of expensive functions through computer experiments (Santner, Williams, and Notz 2003). Since its introduction by Sacks et al. (1989), kriging has been widely used as the interpolation method in this area, but not without any problems. Here we propose an alternative interpolation method and investigate its properties.

A major limitation of kriging is that its computational complexity increases drastically as the number of data points (n) and/or the number of variables (p) increases. This can be easily seen by looking at the kriging predictor:

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})'\hat{\boldsymbol{\beta}} + \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}), \tag{4.1.1}$$

where $\hat{\boldsymbol{\beta}} = (\boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{y}$. Here $\boldsymbol{f}(\boldsymbol{x}) = (f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))'$ is a vector of known functions in $\boldsymbol{x}$, $\boldsymbol{F}$ is the $n \times (m+1)$ regression model matrix, $\boldsymbol{R}$ is the $n \times n$ correlation matrix whose ijth element is $R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\theta})$, and $\boldsymbol{r}(\boldsymbol{x})$ is the $n \times 1$ correlation vector whose ith element is $R(\boldsymbol{x} - \boldsymbol{x}_i; \boldsymbol{\theta})$, where $R(\boldsymbol{h}; \boldsymbol{\theta})$ is the correlation function and $\boldsymbol{\theta}$ is a set of unknown parameters. It can be seen that kriging requires inversion of the $\boldsymbol{R}$ matrix, which becomes difficult as n increases. Furthermore, the computational complexity increases with p. To see this, consider the commonly used Gaussian product correlation function in computer experiments literature:

$$R(\boldsymbol{h}; \boldsymbol{\theta}) = \exp\{-\sum_{i=i}^{p} \theta_i h_i^2\}. \tag{4.1.2}$$

Note that isotropic correlation functions such as the Gaussian correlation function with $\theta_1 = \cdots = \theta_p = \theta$ are not used in computer experiments because the impact of each variable on the output tend to vary. The unknown correlation parameters $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)'$ are usually estimated from data using maximum likelihood method, which is equivalent to minimizing

$$n \log(\hat{\sigma}^2) + \log(det(\boldsymbol{R})), \tag{4.1.3}$$

with respect to $\boldsymbol{\theta}$, where $\hat{\sigma}^2 = 1/n(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})$. Because the dimension of $\boldsymbol{\theta}$ increases with p, optimizing (4.1.3) becomes difficult as p increases. Almost all of the optimization algorithms utilize iterative procedures that require many evaluations of (4.1.3) and for each evaluation, the $\boldsymbol{R}$ matrix needs to be inverted, which makes the optimization very complex. Added to this computational complexity, the $\boldsymbol{R}$ matrix becomes ill-conditioned in some regions of $\boldsymbol{\theta}$, which affects the numerical stability and accuracy of the kriging predictor. These computational and numerical issues have been studied in Ababou, Bagtzoglou, and Wood (1994), Davis and Morris (1997), and An and Owen (1999).

The foregoing computational and numerical problems could be reduced if we were to use a method that does not require the inversion of a large matrix. *Inverse distance weighting* (IDW) proposed by Shepard (1968) is one such interpolating method. However, the IDW is not as accurate as the kriging predictor. Hence we introduce some simple modifications to the IDW predictor to improve its prediction performance.

One might wonder if we really need a predictor that works well with large n, because if the computer experiment is not expensive to run then we may not need an approximator (or emulator) of the computer model. This is not true; in reality, even when the computer experiments are cheap, an emulator can be useful. Here are some situations. The computer experiment may be fast but not fast enough to afford a large number of runs. For example, if one run takes only 30 minutes of evaluation (compared to an expensive computer experiment that takes 12 hours/run (Hung, Joseph, and Melkote 2009)), it is still may not possible to conduct more than say, 1000 runs. This situation worsens when the p is also

large, because a large data set can become really sparse in higher dimensions and thus, an emulator will be needed for predictions. Another situation is when we have functional responses. For example, in the same machining simulation experiment conducted by Hung et al. (2009), a residual stress profile is obtained over the depth of cut which contains 376 values. Thus, although only 30 runs could be afforded in their experiment, the total number of observations became $n = 30 \times 376 = 11,280$. This functional response was not analyzed in Hung et al. (2009) due to the computational and numerical problems of kriging. Another situation is when we need to run a Monte Carlo (MC) simulation on the computer model. For examples, in the evaluation of sensitivity indices or in finding robust settings of control factors, such MC simulations are necessary. Having an emulator can facilitate these MC simulations and the subsequent computations.

One of the major reasons for the popularity of kriging is its ability to produce confidence intervals for prediction. Kriging is probably the only interpolation method in the literature that has this capability. To overcome this deficiency of IDW predictor, we develop a cross-validation based method for constructing confidence intervals. In fact, we show that these cross-validation based confidence intervals outperform the kriging confidence intervals.

The article is organized as follows. In Section 4.2, we review the existing IDW predictor and then, in Section 4.3, we propose the improved IDW predictor. Some examples are given in Section 4.4 to compare the computational and prediction performances of IDW and kriging predictors. In Section 4.5, we develop the new confidence intervals for IDW prediction. We then conclude in Section 4.6 with some remarks and future research directions.

4.2 Inverse Distance Weighting

The IDW predictor is a weighted average of the observations given by

$$\hat{y}(\boldsymbol{x}) = \sum_{k=1}^{n} v_k(\boldsymbol{x}) y_k, \tag{4.2.1}$$

where

$$v_k(\boldsymbol{x}) = \frac{w_k(\boldsymbol{x})}{\sum_{i=1}^{n} w_i(\boldsymbol{x})} \tag{4.2.2}$$

and $w_i(\boldsymbol{x})$ is a weighting function inversely proportional to the Euclidean distance $d(\boldsymbol{x}, \boldsymbol{x}_i) = \{\sum_{j=1}^{p}(x_j - x_{i,j})^2\}^{1/2}$ from $\boldsymbol{x}$ to $\boldsymbol{x}_i$. A common choice for the weighting function is

$$w_i(\boldsymbol{x}) = 1/d^2(\boldsymbol{x}, \boldsymbol{x}_i). \tag{4.2.3}$$

Thus, the prediction at $\boldsymbol{x}$ is more influenced by the nearby points than the distant points. It is easy to see that the same predictor can be obtained using the weighting function

$$w_i(\boldsymbol{x}) = \prod_{j \neq i} d^2(\boldsymbol{x}, \boldsymbol{x}_j). \tag{4.2.4}$$

Thus, the two forms of the weighting function (4.2.3) and (4.2.4) are equivalent in terms of prediction. Therefore, although $w_i(\boldsymbol{x})$ in (4.2.3) is not defined at $\boldsymbol{x} = \boldsymbol{x}_i$, the predictor is defined at all $\boldsymbol{x}$ because of (4.2.4). From (4.2.4), we can see that $w_i(\boldsymbol{x}_j) = 0$ for all $j \neq i$. Thus, $v_i(\boldsymbol{x}_j) = 1$ if $j = i$ and 0 otherwise and therefore, the IDW predictor interpolates the data.

Although the method is very simple, it suffers from poor prediction performance. Therefore, many modified versions of IDW have been proposed in the literature. In one version, the y_k values are replaced by locally fitted polynomials, where the weights are obtained using only the neighboring points. It is referred to as Modified Quadratic Shepard in Franke (1979) and Method I in Franke and Nielson (1980). Not only the prediction but also the computational time is greatly improved with this modification. See its implementation in Renka (1988). Further improvement is obtained by replacing the local polynomials with radial basis functions (Lazzaro and Motefusco 2002). Other modifications to IDW include using a probability metric instead of Euclidean distance (Łukaszyk 2004) and Liszka's method (Liszka 1984), both of which aim to incorporate random mechanics into prediction.

The foregoing methods focus on spatial interpolation with two or three variables and therefore are not suitable for high dimensional interpolation such as those encountered in

computer experiments. The complex systems studied in computer experiments differ from the problems in spatial statistics in several aspects:

1. Global trends are common in many systems.

2. There are many variables and they differ drastically in terms of their impact on the response.

In the next section, we propose some modifications to the IDW predictor to address these issues.

4.3 Regression-Based IDW

Let $\mu(\boldsymbol{x};\boldsymbol{\beta})$ be a regression model to capture the global trends of the observations, where $\boldsymbol{\beta}$ is a set of unknown parameters. Then, an improved IDW predictor that takes into account of the global trends is given by

$$\hat{y}(\boldsymbol{x}) = \mu(\boldsymbol{x};\boldsymbol{\beta}) + \sum_{k=1}^{n} v_k(\boldsymbol{x}) e_k, \tag{4.3.1}$$

where $e_k = y_k - \mu(\boldsymbol{x}_k;\boldsymbol{\beta}) = y_k - \mu_k$ and $v_k(\boldsymbol{x})$ is defined as in (4.2.2). The foregoing predictor is still an interpolator because $\hat{y}(\boldsymbol{x}_i) = \mu_i + e_i = y_i$. We call it *regression-based inverse distance weighting* (RIDW) predictor.

Because the impact of the variables on the response can be quite different, we propose to add coefficients in the Euclidean distance measure:

$$d(\boldsymbol{x}, \boldsymbol{x}_i) = \left\{ \sum_{j=1}^{p} \theta_j (x_j - x_{i,j})^2 \right\}^{1/2}, \tag{4.3.2}$$

where θ_j's are unknown nonnegative parameters. This modification allows for different values of θ_j's and thus incorporates the importance of each variable into the predictor.

A third modification is made on the weighting function. Although the choice $w_i(\boldsymbol{x}) = 1/d^2(\boldsymbol{x}, \boldsymbol{x}_i)$ is commonly quoted in the IDW literature, a truncated version of it is found to perform better in many applications (Franke and Nielson 1980). It is given by $w_i(\boldsymbol{x}) = (R_i - d(\boldsymbol{x}, \boldsymbol{x}_i))_+ / (R_i d(\boldsymbol{x}, \boldsymbol{x}_i)^2)$, where the radius of influence R_i is chosen just large enough

to include a specified number of points around $\boldsymbol{x}_i$. However, finding R_i in large data sets is computationally cumbersome. Therefore, we propose a new weighting function

$$w_i(\boldsymbol{x}) = \frac{\exp\{-d^2(\boldsymbol{x}, \boldsymbol{x}_i)\}}{d^2(\boldsymbol{x}, \boldsymbol{x}_i)}. \tag{4.3.3}$$

At small distances it behaves like the original weighting function, whereas at large distances it behaves like the truncated version. Moreover, it is much easier to use than the truncated version since we do not need to calculate an R_i for each i.

We choose a linear regression model for $\mu(\boldsymbol{x}; \boldsymbol{\beta})$ given by

$$\mu(\boldsymbol{x}; \boldsymbol{\beta}) = \beta_0 + \sum_{i=1}^{m} \beta_i f_i(\boldsymbol{x}),$$

where $f_i(\boldsymbol{x})$ are some basis functions selected from a candidate set of functions C. The linear regression model is quite flexible and easy to use in high dimensions with large data sets, but other choices such as nonlinear or nonparametric regression models can also be made depending on the applications. A convenient set of candidate functions is the set of polynomials. Let r be the order of the polynomial. Thus, a first order polynomial contains only the linear effects of the factors, a second order polynomial contains the linear effects, quadratic effects, and two-factor interactions, and so on. The polynomial terms could be orthogonalized as in An and Owen (2001). The number of functions in C of order r is given by

$$M(r) = \sum_{i=1}^{r} \binom{p + i - 1}{i}.$$

A second order polynomial is a good choice in many engineering applications with small datasets. However, as the data size increases, a second order polynomial can become inadequate. Therefore, we increase the order of the polynomial as n increases. There are many choices. To keep the computational complexity low, we chose to increase $M(r)$ at the rate of $\sqrt{n}$. Justification for this choice will be given later. Thus, we choose r such that $M(r) \approx \sqrt{M(2)n}$, i.e., $r = \bar{r}$ or $\bar{r} + 1$ depending on whether $M(\bar{r})$ or $M(\bar{r} + 1)$ is closer to $\sqrt{M(2)n}$.

There are three unknown parts in the proposed predictor (4.3.1): $\{f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})\}$, $\boldsymbol{\beta}$, and $\boldsymbol{\theta}$. The first step is to identify the functions $\{f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})\}$, which can be done using an appropriate variable selection technique in regression analysis. However, we should keep in mind that the basic objective of this work is to develop an interpolation method for use in large n and/or p case and therefore, it is important to choose a computationally efficient variable selection technique. A good choice is the least angle regression (LARS) (Efron et al. 2004) whose computational cost is no more than that of a least squares fit. In the context of engineering applications, it is even better to use the modified version of LARS proposed in Yuan, Joseph, and Lin (2007), because it incorporates the effect heredity principle and thus, more interpretable models can be obtained. Finally, the size of the model (m) is chosen using cross validation methods.

Once the functions are identified, we use least squares method to estimate $\boldsymbol{\beta}$, i.e.,

$$\hat{\boldsymbol{\beta}} = arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{n} \{y_j - \beta_0 - \sum_{i=1}^{m} \beta_i f_i(\boldsymbol{x}_j)\}^2.$$

Then $\boldsymbol{\theta}$ is estimated using cross-validation methods by fixing $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$. Let $\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{\mu}}$ be the residuals from the regression, where $\hat{\boldsymbol{\mu}} = (\mu(\boldsymbol{x}_1; \hat{\boldsymbol{\beta}}), \ldots, \mu(\boldsymbol{x}_n; \hat{\boldsymbol{\beta}}))'$. The data are then randomly grouped into K folds: S_1,..., S_K and the cross-validation errors are computed. The mean squared cross-validation error is given by

$$MSCV_{IDW}(K) = \frac{1}{n} \sum_{i=1}^{K} \sum_{l \in S_i} (e_l - \hat{e}_l^{CV})^2, \tag{4.3.4}$$

where

$$\hat{e}_l^{CV} = \frac{\sum_{j \notin S_i} w_j(\boldsymbol{x}_l) e_j}{\sum_{j \notin S_i} w_j(\boldsymbol{x}_l)}, \quad \text{for } \; l \in S_i.$$

Now, $\boldsymbol{\theta}$ can be estimated by minimizing the $MSCV_{IDW}(K)$.

Note that we use a two-stage estimation procedure for $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. A single stage estimation could improve the results, but at the expense of increased computations. This is not desirable when dealing with large n and/or p problems. Moreover, the examples we have tried so far did not show much improvement with the single stage estimation that is worth enough for the additional computational investment.

The main improvement of the RIDW predictor comes from its regression part. This modification is quite similar to the use of local polynomials in Franke and Nielson (1980). We use a global polynomial instead of the local polynomials. Clearly, local polynomials can provide better prediction, but global polynomial makes the predictor easier to interpret. Moreover, we use an expanded polynomial basis, which grows in size with n, making it more suitable for application in large data sets. The use of polynomials for interpolation is not a new concept. Bates, Giglio, and Wynn (2003) have proposed an efficient algorithm using algebraic theory to identify polynomial interpolators. Different from them, our aim is not to obtain interpolation using the polynomials. The interpolation is easily achieved using the IDW part. In our predictor, polynomials are used only to capture the global trends.

The use of expanded polynomial basis and variable selection can also be done in fitting kriging models, which is similar in spirit to the blind kriging method (Joseph, Hung, and Sudjianto 2008). However, it further increases the computational cost which is already high for kriging. Let s be the size of the candidate set. The computational complexity of fitting the entire solution path of LARS is given by $O((s \wedge n)^2 + ns(s \wedge n))$, where $s \wedge n = \min(s, n)$ (Yuan et al. 2007). Thus, if this approach is adopted for kriging, then the computational complexity becomes $O((s \wedge n)^2 + ns(s \wedge n) + n^3)$, which can be prohibitive as n gets larger. On the other hand, the computational complexity of RIDW predictor is only $O((s \wedge n)^2 + ns(s \wedge n) + n)$. Recall that we choose $s \approx \sqrt{M(2)n} = O(n^{1/2})$, so that the computational complexity for RIDW becomes $O(n + n^2 + n) = O(n^2)$, which is smaller than $O(n^3)$. Of course, other choices of s are acceptable as long as the computational complexity is much less than $O(n^3)$.

RIDW does not share the "optimality" properties of kriging and thus, is not expected to perform better in terms of prediction. However, if the loss of prediction performance is negligible, then RIDW predictor can be used in situations such as high dimensional

problems with large data sets where kriging cannot be easily applied due to the computational/numerical problems. Therefore, we investigate their performances using some examples.

4.4 Examples

4.4.1 A Small-Scale Experiment

First we consider a small n and p example from the literature. Consider the circuit simulator experiment in Sacks et. al. (1989). There are six experimental factors $x_1 \sim x_6$, which are in the region $[-0.5, 0.5]^6$. The experiment contains 32 runs. The details of the experimental design and data can be found in their original paper.

Because $\sqrt{26 \times 32} = 27.9$ is closer to $M(2) = 26$ than $M(3) = 83$, we choose $r = 2$. Thus, the candidate set has 26 functions, which includes the linear main effects, quadratic effects, and two-factor interactions. The modified LARS identified the following variables (using both weak and strong heredity):

$$\text{Step 1: } x_5, \quad \text{Step 2: } x_6, \quad \text{Step 3: } x_3, \quad \text{Step 4: } x_2,$$
$$\text{Step 5: } x_4, \quad \text{Step 6: } x_4x_6, \quad \text{Step 7: } x_1^2, x_1, \quad \text{Step 8: } x_4^2, \ldots.$$

Figure 4.4.1 shows the $MSCV_{LS}(32)$, $MSCV_{LS}(8)$, and R^2-adjusted along the variable selection path, where $MSCV_{LS}(K)$ is the mean squared K-fold cross-validation error computed using only the linear model. Clearly, the model with the first nine variables is sufficient. Thus,

$$\hat{\mu}(\boldsymbol{x}) = -0.679 + 0.661x_5 - 0.548x_6 - 0.689x_3 + 0.424x_2 + 0.274x_4 + 1.171x_4x_6 + 0.174x_1^2 + 0.090x_1 - 1.187x_4^2.$$

Next, by minimizing the $MSCV_{IDW}(32)$, we obtain $\hat{\boldsymbol{\theta}}_{IDW}$=(0.2471, 0.0001, 0.0001, 0.3901, 0.0001, 6.0705). Thus, the RIDW predictor is given by

$$\hat{y}(\boldsymbol{x}) = \hat{\mu}(\boldsymbol{x}) + \frac{\sum_{k=1}^{n} w_k(\boldsymbol{x}) e_k}{\sum_{i=1}^{n} w_i(\boldsymbol{x})},$$

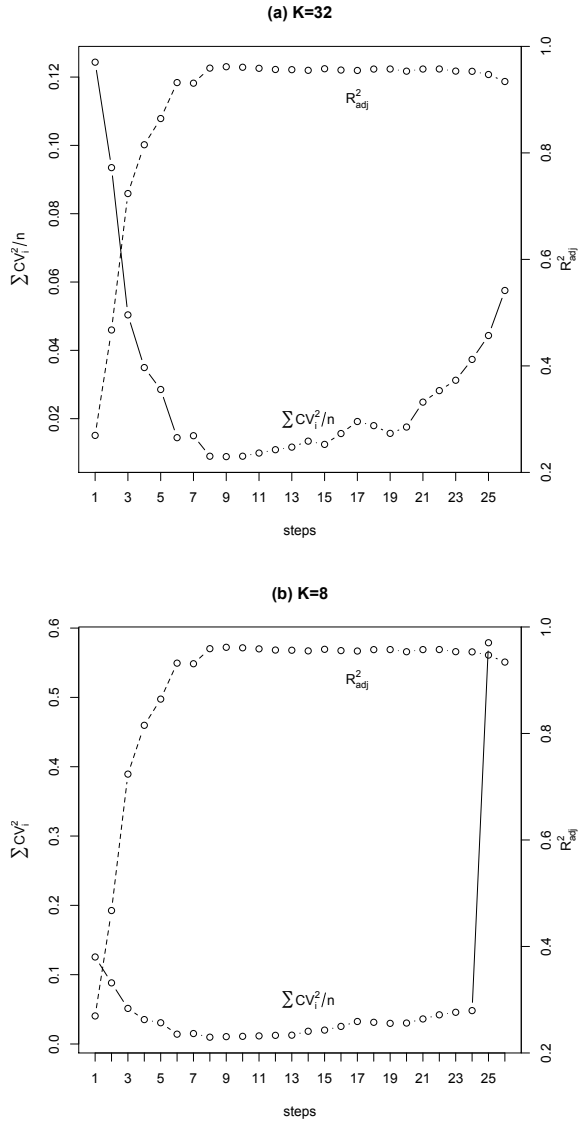


Figure 4.4.1: *K-fold cross-validation error and R^2_{adj} along the variable selection path.*

Table 4.4.1: Circuit-simulator Example: Root Mean-squared cross-validation errors.

K-fold	IDW	Ordinary Kriging	Blind Kriging	RIDW
32-fold	0.369	0.139	0.0761	0.0853
8-fold	0.369	0.142	0.0799	0.0895

where $e_k = y_k - \hat{\mu}(\boldsymbol{x}_k)$ and $w_k(\boldsymbol{x}) = \exp\{\sum_{j=1}^{6} \hat{\theta}_{IDW,j}(x_j - x_{k,j})^2\} / \sum_{j=1}^{6} \hat{\theta}_{IDW,j}(x_j - x_{k,j})^2$.

To compare prediction accuracy, we also construct the kriging predictor. We fit an ordinary kriging model (only a constant in the mean function) with a Gaussian correlation function $R(\boldsymbol{h}) = \exp(-\sum_{i=1}^{6} \theta_i h_i^2)$. The ordinary kriging predictor is given by

$$\hat{y}(\boldsymbol{x})_{OK} = -1.055 + \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{R}^{-1}(\boldsymbol{y} - 1.055 \times \mathbf{1}),$$

where $\mathbf{1}$ is a vector of 1's and $\hat{\boldsymbol{\theta}}_{OK}$=(2.161, 0.553, 0.972, 0.871, 1.024, 4.916).

For a fair comparison, we also consider the blind kriging predictor which uses the same mean function as in the RIDW. The blind kriging predictor is given by

$$\begin{aligned}\hat{y}(\boldsymbol{x})_{BK} = &-0.664 + 0.618x_5 - 0.560x_6 - 0.749x_3 + 0.434x_2 + 0.213x_4 \\ &+ 1.055x_4x_6 - 0.0826x_1^2 + 0.0771x_1 - 1.206x_4^2 + \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}),\end{aligned}$$

where the correlation parameters $\hat{\boldsymbol{\theta}}_{BK}$=(12.02, 1.77, 0.01, 81.29, 0.01, 200.00).

The 32-fold and 8-fold root mean-squared cross-validation errors (RMSCV) for the different methods are shown in Table 4.4.1. Compared to the original IDW method, the improvement obtained by the RIDW method is quite substantial. It has comparable accuracy to that of ordinary and blind kriging. Of course, as done in Sacks et al. (1989), it is possible to improve the performance of kriging by varying the smoothness parameter, α_i, in the power exponential correlation function $R(\boldsymbol{h}) = \exp(-\sum_{i=1}^{6} \theta_i h_i^{\alpha_i})$. However, this doubles the number of unknown correlation parameters, which although is fine with this example, cannot work in large n and p problems.

We have also studied the effect of various changes made to the original IDW predictor. The RMSCV based on the leave-one-out cross validation error (i.e., 32-fold) for the original

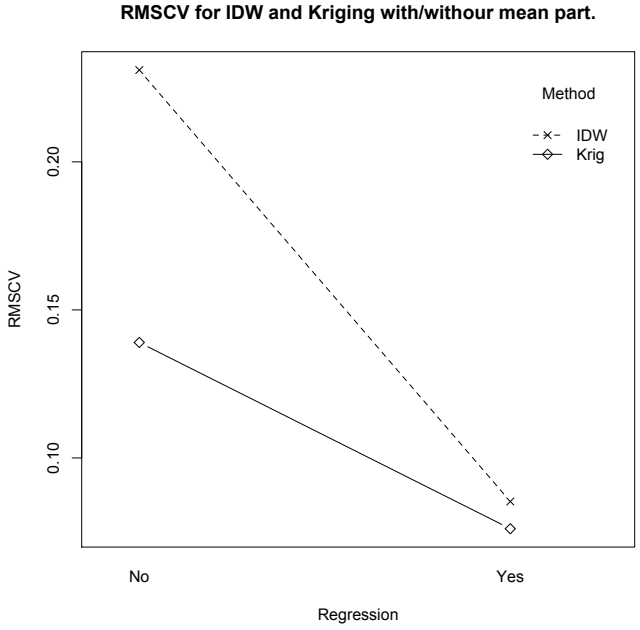


Figure 4.4.2: RMSCV for IDW and kriging with and without regression part.

IDW predictor is 0.369. When we introduced different θ_i's for each variable, the RMSCV is reduced to 0.307. When we changed the weighting function from $w_i(\boldsymbol{x}) = 1/d(\boldsymbol{x}, \boldsymbol{x}_i)^2$ to the exponential weighting function in (4.3.3), the RMSCV is further reduced to 0.231. This is again reduced to 0.0853 by introducing the regression model. Although the amount of improvements depend on the nature of the problem, in general, the changes made to the original IDW predictor seem to help in improving its performance with the regression part playing the major role.

The effect of adding a regression part to the kriging and IDW predictors are shown in Figure 4.4.2. We can see that adding the regression part helps in improving the prediction performance of both the predictors, but the effect on the IDW predictor is more substantial. In other words, kriging predictor is more robust to the regression part than the IDW predictor. Thus, kriging can outperform RIDW when we are not able to identify a good regression model. This is the reason why we chose an expanding polynomial basis for the candidate functions, so that there are higher chances of finding a good regression model.

4.4.2 A Large-Scale Experiment

Now we consider a problem with large n and p. The observations are simulated from the revised Ackley's path function, which is defined as follows

$$f(\boldsymbol{x}) = -a \exp\left\{-b\sqrt{\sum_{i=1}^{p} x_i^2/p}\right\} - \exp\left\{\sum_{i=1}^{p} \cos(cx_i)/p\right\} + a + e^1, \quad \boldsymbol{x} \in [-2, 2]^p,$$

where $a = 2p$, $b = 0.2$, and $c = 2\pi$. In our simulation, we set $p = 10, 25, 50$ and $n = 500, 1000, \ldots, 9000$. We use maximin Latin Hypercube designs (LHD) (McKay, Beckman, and Conover 1979, Morris and Mitchell 1995) obtained using the *lhs* package in R (http://www.cran.r-project.org/web/packages/lhs.) The designs are generated sequentially by augmenting the smaller designs, i.e., a 1000-run design is obtained by adding 500 points to the previously generated 500-run design, and so on. The regression models are selected by using the modified LARS procedure from the candidate set of polynomials as described before. Then, the same regression models are used for both RIDW and blind kriging. Here we use the DACE toolbox in MATLAB for the estimation of the kriging models and their predictions. The unknown coefficients $\boldsymbol{\theta}$ of RIDW are estimated using the same optimization procedure of the DACE toolbox.

The simulation was run on 64-bit two quad-core 2.33 GHz Xeon 5345's CPU each with 4MB cache and 12 GB RAM. For each setting of p, the simulation was terminated when both of the kriging methods failed. To compare the performance of the three methods, we computed root mean squared prediction error (RMSPE) using 500 random LHD points and to compare the computational speed, we recorded the CPU time used by the optimization procedure. The standardized RMSPE, which is the RMSPE divided by the standard deviation of the true function values and the CPU time in seconds are plotted in Figure 4.4.3. We can see that the prediction performance of RIDW is only slightly worse than that of blind kriging, but much better than that of ordinary kriging. On the other hand, RIDW has a clear advantage in the computational speed. The average CPU time for ordinary and blind kriging are respectively 2.47 and 3.17 times more than that of RIDW. Moreover, the saving

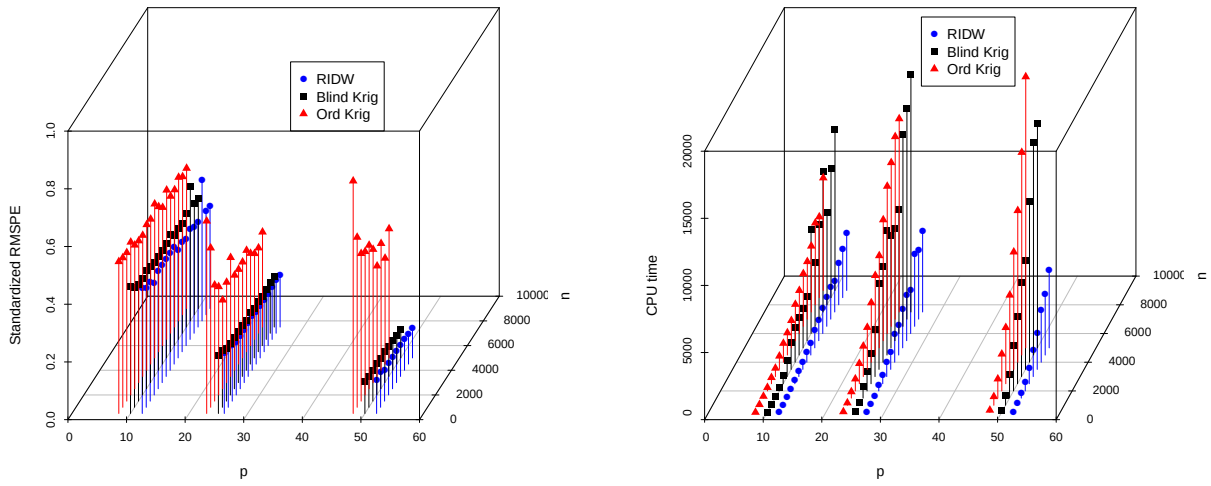


Figure 4.4.3: *Standardized RMSPE (left) and CPU time (right) in simulation.*

in CPU time increases as n and/or p increases.

To save the total simulation time, we have run the optimization using only a single initial value, which was generated randomly from $[0, 1]^p$ for all the three methods. The objective functions for both the kriging and RIDW are multimodal and therefore, the results we obtained need not be the global optimum. Thus, it might be possible to improve the performance of the three methods with a more careful fine tuning of the parameters. However, this cannot be done in a reasonable amount of time. Take for example the case of $n = 3500$ and $p = 50$. The CPU time for the RIDW is 2709.6 seconds and that of blind kriging is 8118 seconds. If we were to use 100 initial values, then the optimization alone will take 3 days for the RIDW and more than 9 days for the blind kriging. In real-life applications, however, more initial values should be used in the optimization procedure.

We also compared RIDW with the Bayesian treed Gaussian process (TGP) model (Gramacy and Lee 2008). Consider the case with $p = 2$ and $n = 500$. The TGP model is fitted using the *tgp* package in R (http://www.cran.r-project.org/web/packages/tgp.) For the RIDW, the optimization is done with five initial starts. The total CPU time consumed by

the RIDW method including the model selection is 0.22 seconds, which is much smaller compared to 1700 seconds taken by the TGP. Another 100 points are generated for prediction. The RMSPE for RIDW is 0.46, whereas that of TGP is 0.92. Thus, in this example, RIDW performs better than the TGP in terms of both prediction accuracy and computational speed. The improvement in prediction accuracy could be attributed to the smooth test function for which the tree partitioning is not very efficient.

4.5 Confidence Interval

In many situations it is useful to report a confidence interval around the predictions. Kriging has a major advantage over the other interpolation methods on this aspect. Its stochastic formulation automatically provides a confidence interval. The kriging $(1-\alpha)$ confidence interval or prediction interval at a point $\boldsymbol{x}$ is given by (Santner et al. 2003, p. 94)

$$\hat{y}(\boldsymbol{x}) \pm z_{\frac{\alpha}{2}}\hat{\sigma}\sqrt{1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})+\boldsymbol{c}(\boldsymbol{x})'(\boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{c}(\boldsymbol{x})}, \tag{4.5.1}$$

where $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})$. The normal quantile $z_{\alpha/2}$ is based on a Gaussian process assumption of the underlying stochastic process (Currin et al. 1991). However, note that we only have a single realization of the stochastic process. Therefore, we cannot verify the validity of the probability model assumptions and hence, the kriging confidence intervals can not always be trusted.

Because the IDW/RIDW does not have a probability model, it is not straightforward to construct a confidence interval. Therefore we use a heuristic approach to develop confidence intervals. First we develop the confidence intervals for IDW and then extend it to RIDW.

4.5.1 Confidence interval for IDW

We take advantage of the similarity between IDW and kriging to develop the confidence intervals. If the mean part is 0, then the kriging confidence interval becomes:

$$\hat{y}(\boldsymbol{x}) \pm z_{\frac{\alpha}{2}}\hat{\sigma}\sqrt{1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})}. \tag{4.5.2}$$

Here the term $1 - \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})$ pulls the confidence intervals to 0 at the observed $\boldsymbol{x}_i$'s. We call this *shrinkage function*.

Motivated by the form of kriging confidence intervals, we propose the confidence intervals of IDW to be

$$\hat{y}(\boldsymbol{x}) \pm \kappa_\alpha \sqrt{\delta(\boldsymbol{x})S(\boldsymbol{x})}, \tag{4.5.3}$$

where $S(\boldsymbol{x})$ is a shrinkage function, $\delta(\boldsymbol{x})$ is a variance function to capture the variability of errors, and κ_α is a scaling constant to obtain the required $(1 - \alpha)$ probability coverage.

Let $\boldsymbol{v}(\boldsymbol{x}) = (v_1(\boldsymbol{x}), \ldots, v_n(\boldsymbol{x}))'$, where $v_i(\boldsymbol{x})$ is defined in (4.2.2). Noting that $\boldsymbol{r}(\boldsymbol{x})$ and $\boldsymbol{v}(\boldsymbol{x})$ play a similar role, we take the shrinkage function to be

$$S(\boldsymbol{x}) = 1 - \boldsymbol{v}(\boldsymbol{x})'\boldsymbol{v}(\boldsymbol{x}).$$

It is easy to see that $S(\boldsymbol{x}_i) = 0$ for all $i = 1, \ldots, n$.

The variance function $\delta(\boldsymbol{x})$ is a constant in the kriging confidence intervals. This is a very restrictive assumption and was needed only to satisfy the stationarity requirements. We can relax this assumption. The leave-one-out cross-validation error $CV_i = y_i - \hat{y}_{-i}(\boldsymbol{x}_i)$ provides an estimate of error in the predictor around $\boldsymbol{x} = \boldsymbol{x}_i$. Therefore, we can interpolate the data $(\boldsymbol{x}_1, CV_1^2), \ldots, (\boldsymbol{x}_n, CV_n^2)$ to understand the extent of variability in the predictor at any given location $\boldsymbol{x}$. For simplicity, we use IDW for interpolation. Thus, the variance function is given by

$$\delta(\boldsymbol{x}) = \sum_{i=1}^{n} v_i(\boldsymbol{x}) CV_i^2.$$

Note that the variance function could have been improved using RIDW interpolation, but here we take the stand that the accuracy of confidence intervals is not as important as the accuracy of the prediction itself and thus a crude approximation to $\delta(\boldsymbol{x})$ is more than enough.

The scaling factor κ_α can be computed as follows. For a given κ_α, we construct the confidence interval at $\boldsymbol{x} = \boldsymbol{x}_i$ after removing $(\boldsymbol{x}_i, y_i)$ from the data set, i.e.,

$$\hat{y}_{-i}(\boldsymbol{x}_i) \pm \kappa_\alpha \sqrt{\delta_{-i}(\boldsymbol{x}_i)\{1 - \tilde{\boldsymbol{v}}(\boldsymbol{x}_i)'\tilde{\boldsymbol{v}}(\boldsymbol{x}_i)\}},$$

where

$$\tilde{v}_j(\boldsymbol{x}_i) = \frac{w_j(\boldsymbol{x}_i)}{\sum_{j\neq i} w_j(\boldsymbol{x}_i)} \text{ and } \delta_{-i}(\boldsymbol{x}_i) = \sum_{k\neq i} \tilde{v}_k(\boldsymbol{x}_i) CV_{k,-i}^2.$$

Here $CV_{k,-i} = y_k - \hat{y}_{k,-i}$ is the cross-validation error at $\boldsymbol{x}_k$ using data without $(\boldsymbol{x}_i, y_i)$. Now we can check if y_i falls within this confidence interval or not. This can be repeated for all $i = 1, \ldots, n$, and we can choose κ_α so that $(1-\alpha)$ fraction of the points fall within the confidence intervals. This can be easily done by setting κ_α to be the upper α sample quantile of

$$\frac{|CV_i|}{\sqrt{\delta_{-i}(\boldsymbol{x}_i)\{1 - \tilde{\boldsymbol{v}}(\boldsymbol{x}_i)'\tilde{\boldsymbol{v}}(\boldsymbol{x}_i)\}}}, i = 1, \ldots, n.$$

For illustration, let $y = \exp(-1.5x)\sin(4\pi x^2)$ for $x \in [0, 3]$ be the underlying true function, which is plotted in solid line in Figure 4.5.1. The curve fluctuates around zero with decreasing amplitude and therefore, the prediction errors are expected to reduce as x increases. We choose 10 equally spaced points in $[0, 3]$. The IDW predictions and 90% confidence intervals based on these 10 points are plotted in the left panel of Figure 4.5.1. We also fitted the ordinary kriging and is plotted in the right panel of the Figure 4.5.1. In this example, we used the penalized likelihood estimation for the correlation parameter (Li and Sudjianto 2005) with the penalty function $p(\theta) = 0.1|\theta|$, which is found to give much better results. The ordinary kriging confidence intervals are computed using the formula

$$\hat{y}(\boldsymbol{x}) \pm z_{\frac{\alpha}{2}}\hat{\sigma}\left\{1 - \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \frac{(1 - \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\mathbf{1})^2}{\mathbf{1}'\boldsymbol{R}^{-1}\mathbf{1}}\right\}^{1/2},$$

and are also plotted in the right panel of Figure 4.5.1. It can be seen that the kriging confidence intervals are roughly the same width throughout the interval $[0, 3]$, which is clearly not adequate for this particular example. On the other hand, the proposed confidence intervals become wider when errors are large and become narrower when errors are small; something that we should expect to see in good confidence intervals.

To check the effect of experimental design on the confidence intervals, we randomly sampled 10 points from $[0, 3]$. The predictions and confidence intervals are plotted in Figure 4.5.2. We can see that the proposed confidence intervals for IDW still work better

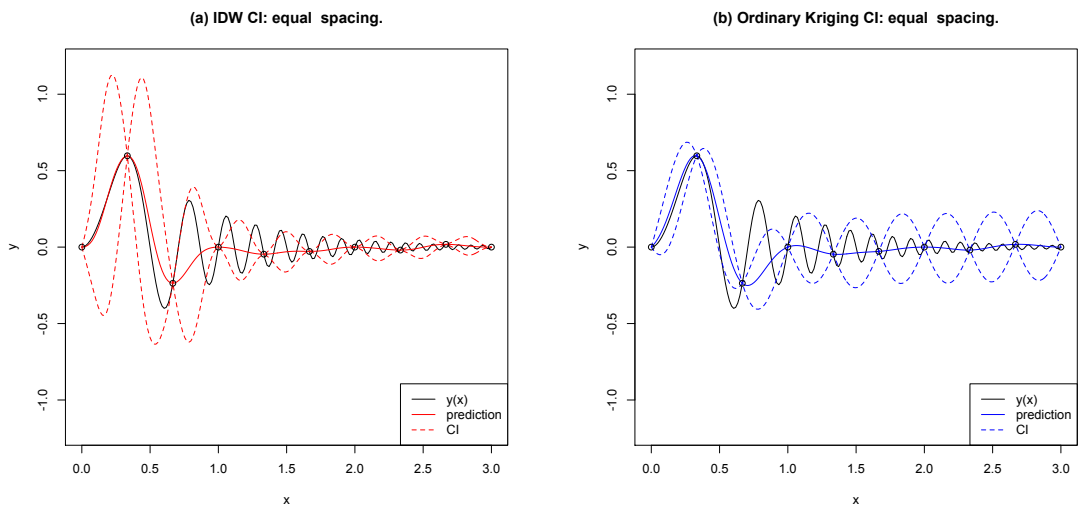


Figure 4.5.1: *Confidence intervals with equally spaced points: (a) kriging (b) IDW.*

than the kriging confidence intervals. Thus, the spacing between points does not seem to affect the performance of the method.

The shrinkage functions for the foregoing two cases are plotted in the top panels of Figure 4.5.3. We can see that they shrink to 0 at the observed locations and expands as the distance increases from the observed locations. The behaviors of kriging and IDW shrinkage functions are quite similar when the points are sampled at equal spacing. Note the behavior looks similar throughout the interval [0, 3] because the shrinkage functions depend only on x and not on $\boldsymbol{y}$. However, drastic changes are observed between the two shrinkage functions when the samples are not equally spaced. In some areas kriging gives more shrinkage and in some other areas IDW gives more shrinkage. However, again they are not dependent on the prediction errors and therefore, it is difficult to say which one is better. On the other hand, as shown in the middle panels of Figure 4.5.3, the variance function $\delta(x)$ captures the prediction errors. We can see that for IDW, the variance function decreases as x increases, whereas for kriging, it is a constant. Clearly, the IDW variance function captures the true behavior. A boxplot of the scaled cross-validation errors are shown in the bottom panels of Figure 4.5.3. We can see that the proposed approach chooses the scaling constant

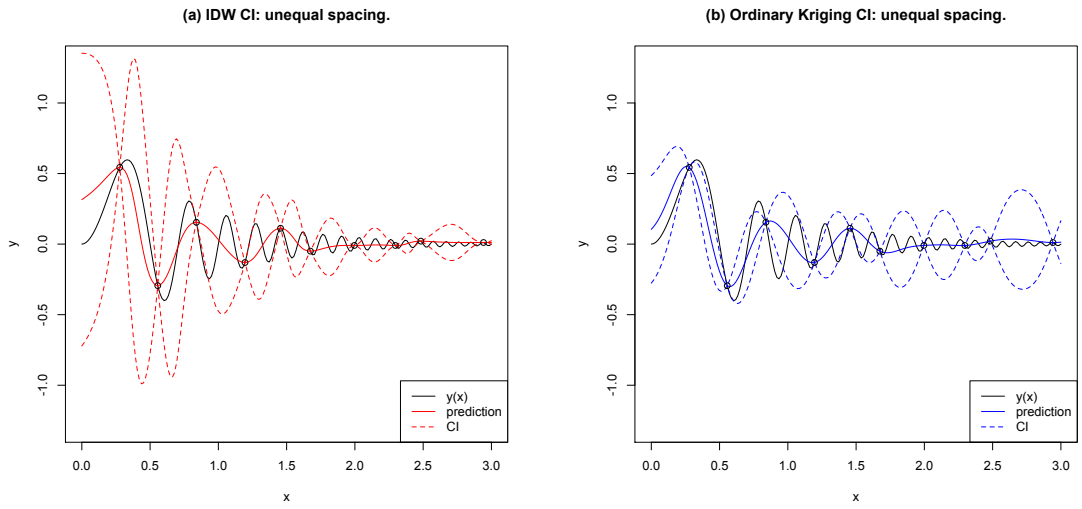


Figure 4.5.2: *Confidence intervals with unequally spaced points: (a) kriging (b) IDW.*

κ_α based on the observed prediction errors, whereas, in kriging, the scaling constant is chosen based on the standard normal critical value. The normal distribution assumption made in kriging is purely for mathematical convenience and thus, the critical value based on it can be inaccurate.

To compare the performance of the confidence intervals of IDW and kriging, we computed the coverage probabilities by randomizing the designs. That is, we generated 10 points randomly from $[0, 3]$ and constructed the confidence intervals for both the predictors. Then, we checked if the true function value at a particular x falls within the confidence intervals. This is then repeated 1000 times and the proportion of the times the point falls within the confidence intervals is noted down. This is plotted in Figure 4.5.4. We can see that the proposed method for IDW achieves the required 90% coverage approximately throughout the interval $[0, 3]$, whereas the kriging confidence interval has far less coverage than 90%. The coverage of kriging is poor on the left side of the figure where the function is more wiggly.

We should point out that modifications can be made on the kriging confidence intervals to improve the coverage. Yamamoto (2000) has proposed a local variance estimator for

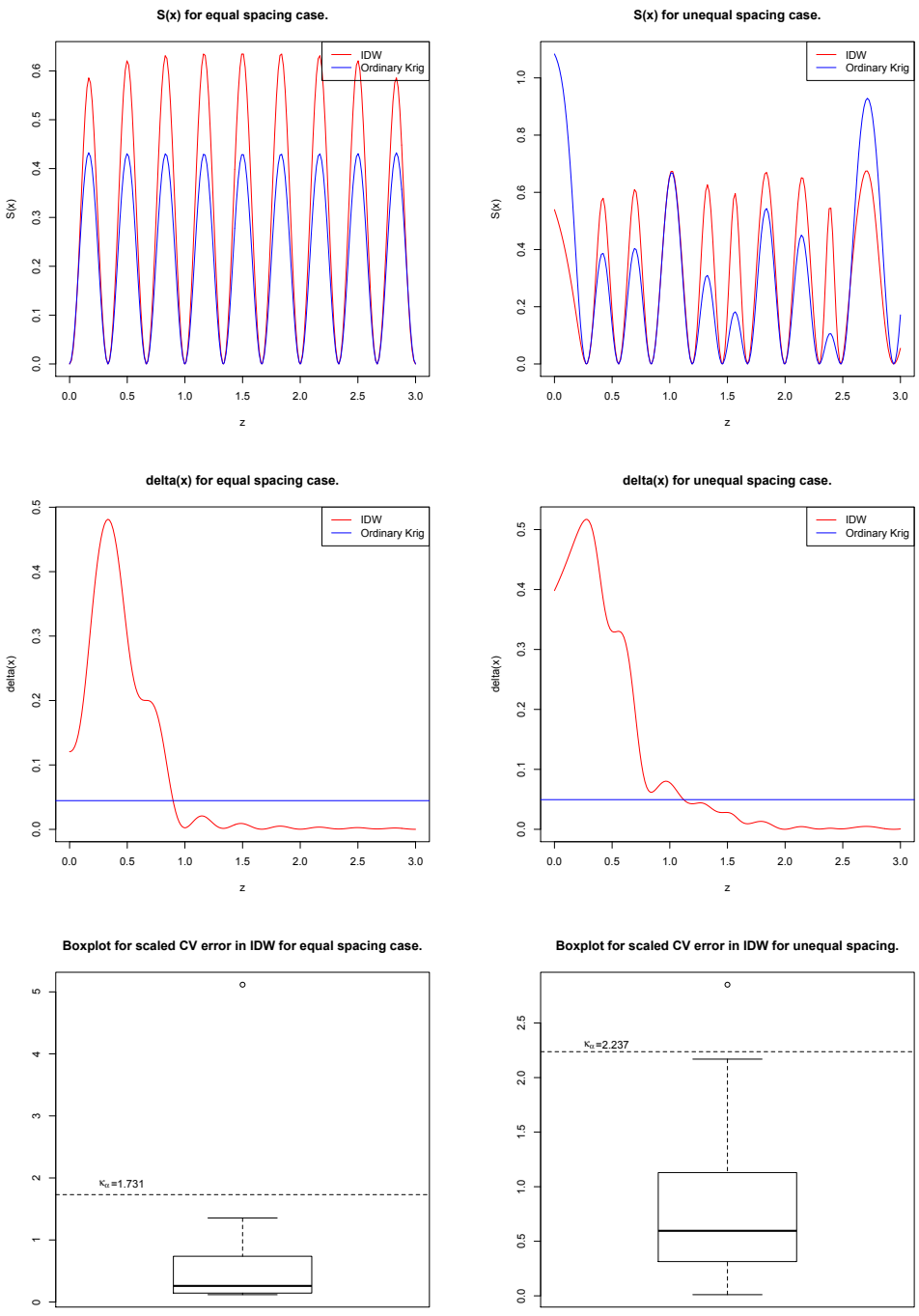


Figure 4.5.3: *Shrinkage functions, variance functions, and scaling constants.*

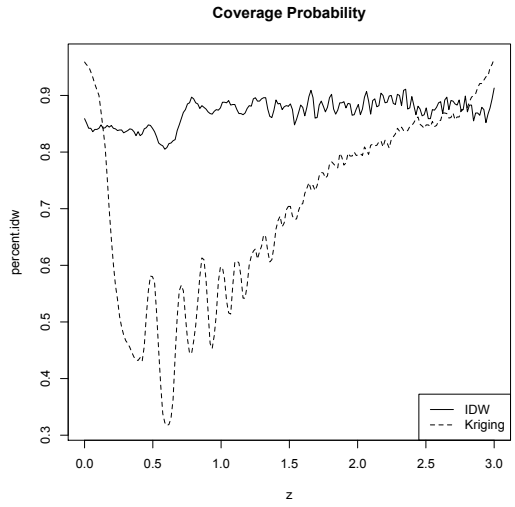


Figure 4.5.4: *Coverage Probability*

ordinary kriging. The Treed Gaussian process approach of Gramacy and Lee (2008) can also improve the coverage, because the space can be partitioned into different regions with the response in each region having different variance. Another approach is to transform the response so that the local variance becomes approximately constant throughout the region. A logarithmic transform of the response in this example will work. However, the transformation approach is not general and is effective only in limited situations. Moreover, as the dimension increases, identifying the right transformation will be difficult, if not impossible.

4.5.2 Confidence interval for RIDW

Based on (4.5.1), we choose the shrinkage function to be

$$S(\boldsymbol{x}) = 1 - \boldsymbol{v}(\boldsymbol{x})'\boldsymbol{v}(\boldsymbol{x}) + \boldsymbol{c}(\boldsymbol{x})'(\boldsymbol{F}'\boldsymbol{F})^{-1}\boldsymbol{c}(\boldsymbol{x}),$$

where $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{F}'\boldsymbol{v}(\boldsymbol{x})$. It is easy to verify that $S(\boldsymbol{x}_i) = 0$ for $i = 1, \ldots, n$. The variance function $\delta(\boldsymbol{x})$ is the same as in the IDW confidence interval except that now CV_i is the leave-one-out cross-validation error for the RIDW predictor. We also need to modify

κ_α accordingly. It is set to be the upper α sample quantile of

$$\frac{|CV_i|}{\sqrt{\delta_{-i}(\boldsymbol{x}_i)\{1-\tilde{\boldsymbol{v}}(\boldsymbol{x}_i)'\tilde{\boldsymbol{v}}(\boldsymbol{x}_i)+\boldsymbol{c}'_{-i}(\boldsymbol{F}'_{-i}\boldsymbol{F}_{-i})^{-1}\boldsymbol{c}_{-i}\}}}. \tag{4.5.4}$$

Here $\boldsymbol{F}_{-i}$ is $\boldsymbol{F}$ without the ith row $\boldsymbol{f}'_i$ and $\boldsymbol{c}_{-i} = \boldsymbol{f}_i - \boldsymbol{F}'_{-i}\tilde{\boldsymbol{v}}(\boldsymbol{x}_i)$. The inversion $(\boldsymbol{F}'_{-i}\boldsymbol{F}_{-i})^{-1}$ can be easily updated by using the formula

$$(\boldsymbol{F}'_{-i}\boldsymbol{F}_{-i})^{-1} = (\boldsymbol{F}'\boldsymbol{F})^{-1} + \frac{(\boldsymbol{F}'\boldsymbol{F})^{-1}\boldsymbol{f}_i\boldsymbol{f}'_i(\boldsymbol{F}'\boldsymbol{F})^{-1}}{1-\boldsymbol{f}'_i(\boldsymbol{F}'\boldsymbol{F})^{-1}\boldsymbol{f}_i}. \tag{4.5.5}$$

The computation of CV_i and $CV_{k,-i}$ can become tedious when n is large. Fortunately, we can derive a short cut formula (4.5.6) to compute them. Let $\boldsymbol{e}$ be the residuals from the estimated regression model, $\boldsymbol{e}_{-i}$ be the vector $\boldsymbol{e}$ excluding e_i, $\tilde{\boldsymbol{v}}_i$ be the vector containing the $n-1$ elements $\tilde{v}_k(\boldsymbol{x}_i)$ for $k \neq i$, and $\boldsymbol{H} = \boldsymbol{F}(\boldsymbol{F}'\boldsymbol{F})^{-1}\boldsymbol{F}'$ be the hat matrix.

Proposition 4.5.1. *The leave-one-out cross-validation error can be computed by*

$$CV_i = \frac{e_i}{1-h_i}(1-\tilde{\boldsymbol{v}}'_i\boldsymbol{b}_i) - \tilde{\boldsymbol{v}}'_i\boldsymbol{e}_{-i}, \tag{4.5.6}$$

where h_i is the ith diagonal entry of $\boldsymbol{H}$ and $\boldsymbol{b}_i$ is the ith column of $\boldsymbol{H}$ without h_i.

The proof is given in the Appendix 4.7. The computation of $CV_{k,-i}$ can also be made simple using Proposition 1. To compute $CV_{k,-i}$, use (4.5.6) to the data set after excluding point $(\boldsymbol{x}_i, y_i)$. The key is to compute the hat matrix for the reduced data set, which is trivial. Denote the hat matrix for data set without $(\boldsymbol{x}_i, y_i)$ as $\boldsymbol{H}_{-i}$. Its updating formula can be found in the proof of Proposition 1.

Consider the following test function used by Currin et al. (1991):

$$y(x_1,x_2) = \{1-\exp(-.5/x_2)\}\frac{2300x_1^3+1900x_1^2+2092x_1+60}{100x_1^3+500x_1^2+4x_1+20}, \boldsymbol{x}\in[0,1]^2.$$

We generated data using a 36-run grid design in $[0,1]^2$. The surface of $y(x_1, x_2)$ is shown in Figure 4.5.5. The mean model:

$$\mu(\boldsymbol{x},\boldsymbol{\beta}) = \beta_0+\beta_1x_2+\beta_2x_1+\beta_3x_1^3x_2+\beta_4x_1^4+\beta_5x_1^3+\beta_6x_1^2x_2+\beta_7x_2^4$$
$$+\beta_8x_1^2+\beta_9x_1x_2+\beta_{10}x_2^2+\beta_{11}x_1x_2^3+\beta_{12}x_1^2x_2^2+\beta_{13}x_2^3,$$

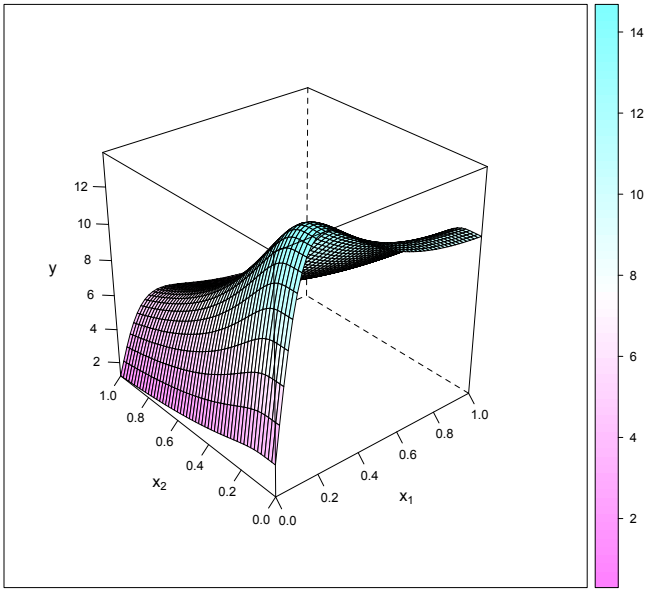


Figure 4.5.5: *The surface of the test function* $y(x_1, x_2)$.

is selected using the LARS method from the candidate set of fourth order polynomials. Then we fit the RIDW and the blind kriging using the same $\mu(\boldsymbol{x}, \boldsymbol{\beta})$. The predicted surfaces are shown in Figure 4.5.6. Clearly, both methods provide similar fitting. To compute the coverage probabilities, as before, we ran 1000 simulations and in each simulation, a random 36-run LHD is generated and the 90% confidence intervals were computed. Figure 4.5.7 shows the coverage probability, which is the proportion of the times the true value fell within the confidence intervals. In the region $[0, 0.5] \times [0, 0.5]$, the average coverage probability of RIDW is 0.786 and that of kriging is 0.515. Whereas, in the region $[0.5, 1] \times [0.5, 1]$, the average coverage probabilities are 0.836 and 0.743 for RIDW and kriging, respectively. Thus, in the regions where the surface is steep or bumpy, kriging confidence intervals have far less coverage than the target 90%, whereas RIDW confidence intervals maintain a much better coverage throughout the experimental region.

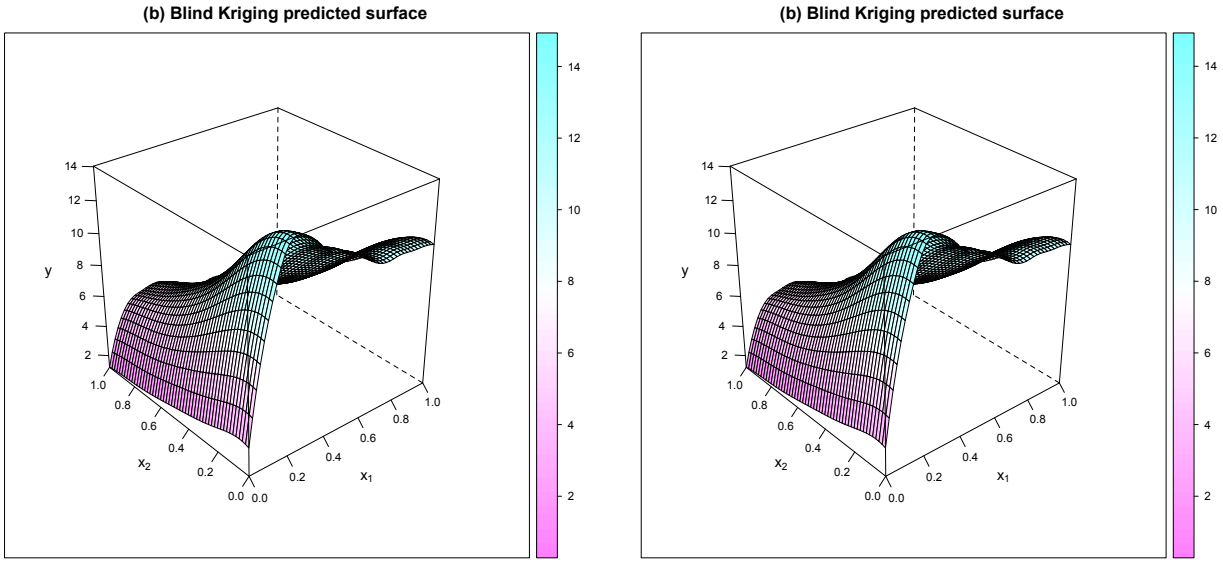


Figure 4.5.6: *Prediction (a) RIDW; (b) Blind Kriging.*

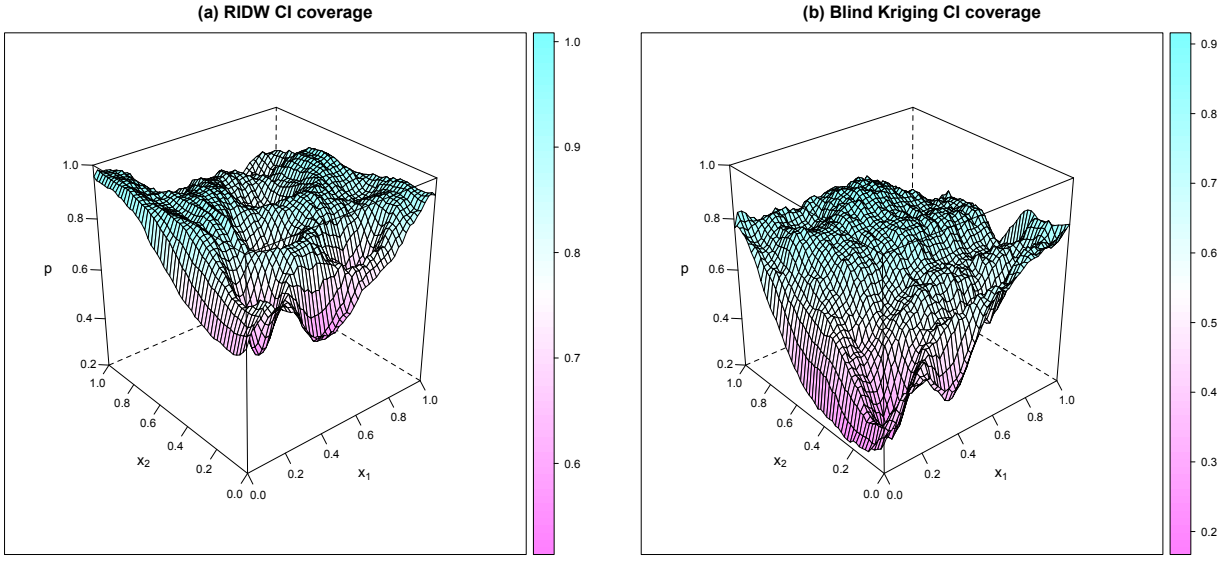


Figure 4.5.7: *Confidence interval coverage (a) RIDW; (b) Blind Kriging.*

4.6 Conclusions

In this article we have shown that by integrating IDW with the linear regression model, a useful method for multivariate interpolation can be obtained. Its advantages are demonstrated through many examples. It is shown that the new RIDW predictor has comparable prediction accuracy to that of kriging and is computationally less demanding. This gives an advantage for RIDW in dealing with large n and/or p problems.

We have also introduced a heuristic method for constructing confidence intervals for RIDW. This overcomes one of its disadvantages for use in computer experiments. Interestingly, the RIDW confidence intervals are shown to perform much better than the kriging confidence intervals. This is mainly because of the adaptive variance function and scaling factor used in the construction of the confidence intervals. These concepts can be extended to kriging confidence intervals as well to improve their performance. However, the extension requires a complete rethinking of the kriging model assumptions. Moreover, its implementation can be computationally challenging in large n and p problems because of the difficulty in computing the cross validation errors. We therefore leave this topic for future research.

4.7 Appendix: Proof of Proposition 1

Let $\hat{y}_{-i}^{LS}$ and $\hat{y}_{-i}$ be the predictions at $\boldsymbol{x}_i$ using the linear regression model and RIDW fitted using data without $\boldsymbol{x}_i$, respectively. Rewrite CV_i as:

$$CV_i = y_i - \hat{y}_{-i} = y_i - \hat{y}_{-i}^{LS} + \hat{y}_{-i}^{LS} - \hat{y}_{-i}. \tag{4.7.1}$$

It is well known from linear regression literature that

$$y_i - \hat{y}_{-i}^{LS} = \frac{e_i}{1 - h_i}. \tag{4.7.2}$$

So we only need to compute $\hat{y}_{-i}^{LS} - \hat{y}_{-i}$. First, note that

$$\hat{y}_{-i} = \hat{y}_{-i}^{LS} + \frac{\sum_{k \neq i} w_k(\boldsymbol{x}_i) e_k^{-i}}{\sum_{k \neq i} w_k(\boldsymbol{x}_i)},$$

where $\boldsymbol{e}^{-i} = \{\boldsymbol{I}_{n-1} - \boldsymbol{F}_{-i}(\boldsymbol{F}'_{-i}\boldsymbol{F}_{-i})^{-1}\boldsymbol{F}'_{-i}\}\boldsymbol{y}_{-i}$, $\boldsymbol{F}_{-i}$ is $\boldsymbol{F}$ without the ith row, and $\boldsymbol{y}_{-i}$ is $\boldsymbol{y}$ without y_i. Thus $\boldsymbol{e}^{-i}$ is a vector of length $n-1$. Now we show how to compute $\boldsymbol{e}^{-i}$.

Partition the hat matrix $\boldsymbol{H}$ into submatrices:

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{A}_i, & \boldsymbol{b}_i \\ \boldsymbol{b}'_i, & h_i \end{pmatrix}.$$

Denote $\boldsymbol{H}_{-i} = \boldsymbol{F}_{-i}(\boldsymbol{F}'_{-i}\boldsymbol{F}_{-i})^{-1}\boldsymbol{F}'_{-i}$. It can be easily shown that

$$\boldsymbol{H}_{-i} = \boldsymbol{A}_i + \frac{\boldsymbol{b}_i\boldsymbol{b}'_i}{1-h_i}.$$

Besides, $\boldsymbol{e}_{-i} = (\boldsymbol{I}_{n-1} - \boldsymbol{A}_i)\boldsymbol{y}_{-i} - y_i\boldsymbol{b}_i$ and $e_i = -\boldsymbol{b}'_i\boldsymbol{y}_{-i} + (1-h_i)y_i$. Therefore, we have

$$\boldsymbol{e}^{-i} = \boldsymbol{e}_{-i} + \frac{e_i}{1-h_i}\boldsymbol{b}_i.$$

Thus,

$$\hat{y}_{-i} - \hat{y}^{LS}_{-i} = \tilde{\boldsymbol{v}}'_i\boldsymbol{e}_{-i} + \frac{e_i}{1-h_i}\tilde{\boldsymbol{v}}'_i\boldsymbol{b}_i. \tag{4.7.3}$$

Substituting (4.7.2) and (4.7.3) in (4.7.1), we obtain the desired result.

CHAPTER V

KERNEL SUM REGRESSION AND INTERPOLATION

5.1 Introduction

Kernel smoothing technique is a very simple yet useful approach to find structure in data sets without imposing a parametric model. There have been many methods and theories studied in this field, including simple kernel estimators and local polynomial modeling (Fan and Gijbels (1996)). Among them, Nadaraya-Watson estimator (Nadaraya (1964) and Watson (1964)) was one of the earliest kernel regression methods introduced. It has a much simpler prediction form compared to its counterparts such as Gasser-Müller estimator (Gasser and Müller (1984)) and local polynomial regression.

Consider the following common regression setting. Let $(\boldsymbol{x}_i, y_i)$ with $i = 1, \ldots, n$ be a set of vectors in $\mathbb{R}^{d+1}$. The vector $\boldsymbol{x}_i \in \Omega \subset \mathbb{R}^d$ are design points, and $\Omega \subset \mathbb{R}^d$ is the design region. The corresponding univariate response at $\boldsymbol{x}_i$ is $y_i \in \mathbb{R}$. Then the Nadaraya-Watson (NW) estimator is,

$$\hat{y}_{NW}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i) y_i}{\sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i)}. \tag{5.1.1}$$

Here $K(\cdot, \cdot) : \Omega \times \Omega \to \mathbb{R}$ is the kernel function, and it usually contains some unknown bandwidth or smoothing parameters. For instance, the common Gaussian kernel is defined as

$$K_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(- \sum_{i=1}^{d} \theta_i (x_{1,i} - x_{2,i})^2 \right). \tag{5.1.2}$$

Compared with other more advanced nonparametric modeling techniques such as splines, although kernel regression methods in general have simpler prediction forms and much less computation, they are not powerful in modeling complex data structure, especially with high dimensional input. There have been numerous works in the literature dedicated to improve the kernel regression. In this paper, our motivation is to take advantage of the

simplicity of the NW estimator and build a new estimator by iteratively performing NW estimator on the residuals from previous regression. As simple and intuitive as this idea is, it leads to an interesting discovery of a new multivariate interpolation method.

The rest of the sections are arranged as follows. In Section 5.2, we proposes the idea of iterative implementation of kernel regression, and develop the algorithm to choose number of regressions and compute the optimal bandwidth parameter based on the generalized cross-validation criterion. We name the proposed method kernel sum regression. An algorithm is proposed to select the optimal number of regressions N and bandwidth parameters based on the generalized cross-validation criterion. In Section 5.3, we show the interesting discovery that if infinite number of kernel regressions are used, the kernel sum regression methods can achieve interpolation, and we call it kernel sum interpolation. Some interesting connections between kernel sun interpolation and other interpolation methods are illustrated, and the kernel sum interpolation is more robust to the choice of bandwidth parameter compared to the other methods. The chapter is concluded in Section 5.4. All the proofs are included in Appendix 5.5.

5.2 Kernel Sum Regression

5.2.1 The prediction form

Joseph and Kang (2010) propose the regression-based inverse distance weighting interpolator (RIDW), which combines the regression with the original inverse distance weighting (IDW) method. In the RIDW interpolator, regression model is used to capture the nonlinear variation and IDW is applied to the residuals of the regression to achieve interpolation and meanwhile to improve the prediction accuracy. To improve the kernel regression, it is very intuitive to apply the similar idea here, i.e., applying another kernel regression on the residuals of the first kernel regression. Let $\boldsymbol{e}_1$ be the residuals of the NW estimator, i.e., $\boldsymbol{e}_1 = \boldsymbol{y} - \hat{\boldsymbol{y}}_1$, where $\boldsymbol{y}$ is the vector of responses and $\hat{\boldsymbol{y}}_1$ is vector of the estimated responses using (5.1.1). Improve the estimator by applying the NW estimator on $\boldsymbol{e}_1$ as well, and the

new estimator is defined:

$$\hat{y}_2(\boldsymbol{x}) = \frac{\sum_{i=1}^n K_1(\boldsymbol{x}, \boldsymbol{x}_i) y_i}{\sum_{i=1}^n K_1(\boldsymbol{x}, \boldsymbol{x}_i)} + \frac{\sum_{i=1}^n K_2(\boldsymbol{x}, \boldsymbol{x}_i) e_{1,i}}{\sum_{i=1}^n K_2(\boldsymbol{x}, \boldsymbol{x}_i)}.$$

The two kernels K_1 and K_2 can be the exact same kernel functions, or the same kernel functions with different bandwidth parameters, or different kernel functions with different bandwidth parameters. Using the same idea, we can repeatedly apply the NW estimator to the residuals. Suppose we use N estimators, and the new estimator is defined as:

$$\hat{y}_N(\boldsymbol{x}) = \frac{\sum_{i=1}^n K_1(\boldsymbol{x}, \boldsymbol{x}_i) y_i}{\sum_{i=1}^n K_1(\boldsymbol{x}, \boldsymbol{x}_i)} + \frac{\sum_{i=1}^n K_2(\boldsymbol{x}, \boldsymbol{x}_i) e_{1,i}}{\sum_{i=1}^n K_2(\boldsymbol{x}, \boldsymbol{x}_i)} + \cdots + \frac{\sum_{i=1}^n K_N(\boldsymbol{x}, \boldsymbol{x}_i) e_{N-1,i}}{\sum_{i=1}^n K_N(\boldsymbol{x}, \boldsymbol{x}_i)}. \tag{5.2.1}$$

The residuals $\boldsymbol{e}_j = \boldsymbol{y} - \hat{\boldsymbol{y}}_j$ are the residuals from kernel sum regression $\hat{y}_j(\cdot)$. Since the new estimator is a sum of series of kernel regression models, we call it *kernel sum regression* (KSR).

Written in matrix form, the kernel sum regression (5.2.1) is

$$\hat{y}_N(\boldsymbol{x}) = \boldsymbol{u}_1(\boldsymbol{x})\boldsymbol{y} + \boldsymbol{u}_2(\boldsymbol{x})\boldsymbol{e}_1 + \cdots + \boldsymbol{u}_N(\boldsymbol{x})\boldsymbol{e}_{N-1}, \tag{5.2.2}$$

where the weight vector is

$$\boldsymbol{u}_j(\boldsymbol{x}) = \left(\frac{K_j(\boldsymbol{x}, \boldsymbol{x}_1)}{\sum_{i=1}^n K_j(\boldsymbol{x}, \boldsymbol{x}_i)}, \ldots, \frac{K_j(\boldsymbol{x}, \boldsymbol{x}_n)}{\sum_{i=1}^n K_j(\boldsymbol{x}, \boldsymbol{x}_i)} \right)'. \tag{5.2.3}$$

Let the weight matrix be $\boldsymbol{U}_i = \boldsymbol{S}_i^{-1}\boldsymbol{K}_i$, where $(\boldsymbol{K}_i)_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s)$, and $\boldsymbol{S}_i$ is the diagonal matrix with $(\boldsymbol{S}_i)_{r,r} = \sum_{j=1}^n K_i(\boldsymbol{x}_j, \boldsymbol{x}_r)$. The residuals of the kernel sum regression $\hat{y}_k(\cdot)$ are

$$\begin{aligned}
\boldsymbol{e}_1 &= \boldsymbol{y} - \boldsymbol{U}_1\boldsymbol{y} = (\boldsymbol{I}_n - \boldsymbol{U}_1)\boldsymbol{y} \\
\boldsymbol{e}_2 &= \boldsymbol{y} - \hat{\boldsymbol{y}}_1 - \boldsymbol{U}_2\boldsymbol{e}_1 = \boldsymbol{e}_1 - \boldsymbol{U}_2\boldsymbol{e}_1 = (\boldsymbol{I}_n - \boldsymbol{U}_2)(\boldsymbol{I}_n - \boldsymbol{U}_1)\boldsymbol{y} = \prod_{i=1}^{2}(\boldsymbol{I}_n - \boldsymbol{U}_i)\boldsymbol{y} \\
&\vdots \\
\boldsymbol{e}_k &= \boldsymbol{y} - \hat{\boldsymbol{y}}_{k-1} - \boldsymbol{U}_k\boldsymbol{e}_{k-1} = \boldsymbol{e}_{k-1} - \boldsymbol{U}_k\boldsymbol{e}_{k-1} = \prod_{i=1}^{k}(\boldsymbol{I}_n - \boldsymbol{U}_i)\boldsymbol{y}
\end{aligned}$$

Therefore, the kernel sum regression estimator can be written in simple matrix form

$$\hat{y}_N(\boldsymbol{x}) = \left(\sum_{i=1}^{N} \boldsymbol{u}_i(\boldsymbol{x})' \prod_{j=0}^{i-1}(\boldsymbol{I}_n - \boldsymbol{U}_j) \right) \boldsymbol{y}. \tag{5.2.4}$$

If we use the same kernel function with the same bandwidth parameters, (5.2.4) becomes

$$\hat{y}_N(\boldsymbol{x}) = \left(\boldsymbol{u}(\boldsymbol{x})' \sum_{i=0}^{N} (\boldsymbol{I}_n - \boldsymbol{U})^i \right) \boldsymbol{y}. \tag{5.2.5}$$

Throughout the paper, we define the matrix product $\prod_{i=1}^{n} \boldsymbol{A}_i = \boldsymbol{A}_n \boldsymbol{A}_{n-1} \ldots \boldsymbol{A}_2 \boldsymbol{A}_1$, $\boldsymbol{U}_0 = \boldsymbol{0}$, and $\boldsymbol{A}^0 = \boldsymbol{I}$ for any matrix $\boldsymbol{A}$.

5.2.2 Estimation

In the kernel sum regression model, there are two types of unknown parameters, the bandwidth parameters and the number of regressions N. As in the other kernel regression models, the unknown bandwidth parameters need to be specified. The optimal bandwidth values would achieve a balance between smoothness of the fitted curve and the prediction accuracy. But kernel sum regression is more complicated, since each kernel regression can use different kernel function and different bandwidth parameters, thus there are N unknown set of parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_N$. For the kernel sum regression, the number of regression N plays the similar role as the bandwidth parameters. If $N = 1$, then the kernel sum regression is simply the NW estimator. In the Section 3 we will show that as N goes to infinity, the kernel sum regression model converges to an interpolator. Therefore, an ideal value of N should also achieve a balance between the smoothness and accuracy. In fact, it is a balance between variation and bias of the fitted model.

We develop an algorithm to simultaneously select the optimal number of regressions N and the bandwidth parameters. Cross-validation criterion is usually used to select the unknown parameters in most nonparametric models. But it is usually computationally intensive. Instead, we use the generalized cross-validation (GCV) (Golub, Heath, Wahba (1979)), which is an improved version of cross-validation and also frequently used in many nonparametric approaches. If the general form of any linear predictor is written as

$$\hat{\boldsymbol{y}} = \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{y},$$

then the GCV criterion is defined as:

$$GCV(\boldsymbol{\theta}) = \left[n^{-1} tr\left\{\boldsymbol{I}_n - \boldsymbol{H}(\boldsymbol{\theta})\right\}\right]^{-2} \left[n^{-1} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2\right].$$

In the kernel sum regression model, the hat matrix is

$$\boldsymbol{H}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N, N) = \sum_{i=1}^{N} \boldsymbol{U}_i \prod_{j=0}^{i-1} (\boldsymbol{I}_n - \boldsymbol{U}_j).$$

Thus the GCV criterion for kernel sum regression is:

$$GCV(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N, N) = \left[n^{-1} tr\left\{\prod_{i=1}^{N} (\boldsymbol{I}_n - \boldsymbol{U}_i)\right\}\right]^{-2} \left[n^{-1} \sum_{i=1}^{n} (y_i - \hat{y}_N(\boldsymbol{x}_i))^2\right]. \tag{5.2.6}$$

The optimal N and $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$,...,$\boldsymbol{\theta}_N$ are the ones that minimize (5.2.6).

To find the optimal N and bandwidth parameter values, we propose a sequential optimization of the GCV criterion to the simultaneous optimization. There are two reasons. First, in the kernel sum regression, the number of regression N decides the function forms of the predictor, thus it should be chosen before the bandwidth parameter values. Second, the kernel sum regression has a sequential form that the ith regression is applied to the residuals of the $(i-1)$th regression. Therefore, a sequential optimization procedure is more practical and intuitive. In the following, we give the algorithm to return the optimal N and bandwidth parameters.

Algorithm 5.2.1. *The sequential optimization procedure for searching N and* $\boldsymbol{\theta}_i$, $i = 1, \ldots, N$.

Step 0: Decide the input arguments value:

- *The lower and upper bounds for bandwidth parameters:* $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}_+^d$.
- *The threshold value for GCV:* $0 < tol < 1$.
- *The maximum bound for N:* $\max_N \geq 1$.

Step 1: Let $N = 1$*, optimize GCV in* (5.2.6)*, and compute* $\boldsymbol{U}_1$, $\boldsymbol{S} = \boldsymbol{I}_n - \boldsymbol{U}_1$, $\boldsymbol{e} = \boldsymbol{S}\boldsymbol{y}$ *based on the optimal* $\boldsymbol{\theta}_1$*. Update the threshold tolerance value to be* $tol = tol \times GCV$.

Step 2: Increase $N \leftarrow N+1$, optimize the GCV with respect to $\boldsymbol{\theta}_N$,

$$GCV(\boldsymbol{\theta}_N) = n\left[tr\left\{(\boldsymbol{I}_n - \boldsymbol{U}_N)\boldsymbol{S}\right\}\right]^{-2}\left[\boldsymbol{e}'(\boldsymbol{I}_n - \boldsymbol{U}_N')(\boldsymbol{I}_n - \boldsymbol{U}_N)\boldsymbol{e}\right],$$

with updated boundaries $\boldsymbol{a} = \boldsymbol{\theta}_{N-1} \leq \boldsymbol{\theta}_N \leq \boldsymbol{b}$. Then update $\boldsymbol{S} \leftarrow (\boldsymbol{I}_n - \boldsymbol{U}_N)\boldsymbol{S}$, $\boldsymbol{e} \leftarrow (\boldsymbol{I}_n - \boldsymbol{U}_N)\boldsymbol{e}$ based on the optimal solution.

Step 3: Repeat Step 2 and stop at N if the optimal GCV starts to increase ($GCV(N+1) > GCV(N)$), or become stable ($|GCV(N+1) - GCV(N)| < tol$) or $N+1 > \max_N$.

In Algorithm 5.2.1, we lift the lower bounds of the parameter $\boldsymbol{\theta}_i$ as the number of regression increases, and the returned optimal bandwidth should satsify $\boldsymbol{\theta}_1 \leq \boldsymbol{\theta}_2 \leq \cdots \leq \boldsymbol{\theta}_N$. It is a well-recognized fact that as the bandwidth parameter becomes larger, the fitted curve by kernel regression will be closer to the actual observations, farther from the global mean, and thus less smooth. Therefore, as N becomes larger, the Nth kernel regression would try harder to capture the local variation than the global trend, which is exactly what it is supposed to achieve, since the global trend would be captured by the previous kernel regressions.

5.2.3 Examples

In this part we illustrate the use of Algorithm 5.2.1 using two examples. In both of the examples, we use Gaussian kernel function (5.1.2) for KSR, NW, and local linear regression models, and Gaussian correlation function (which is the same as the Gaussian kernel) for Gaussian process (GP) models.

Example 5.2.1. *Motorcycle Data (Schmidt, Mattern and Schüler (1981)): the input variable is the time (in milliseconds) after a simulated impact on motorcycles, and the response variable is the head acceleration (in g) of a test object. There are 133 pairs of observations, which are plotted as black circles in both of the panels in Figure 5.2.3.*

Using the Algorithm 5.2.1, the optimal GCV value starts to increases from $N = 4$, thus the optimal number of regressions should be $N = 3$. We compare the KSR fitting with

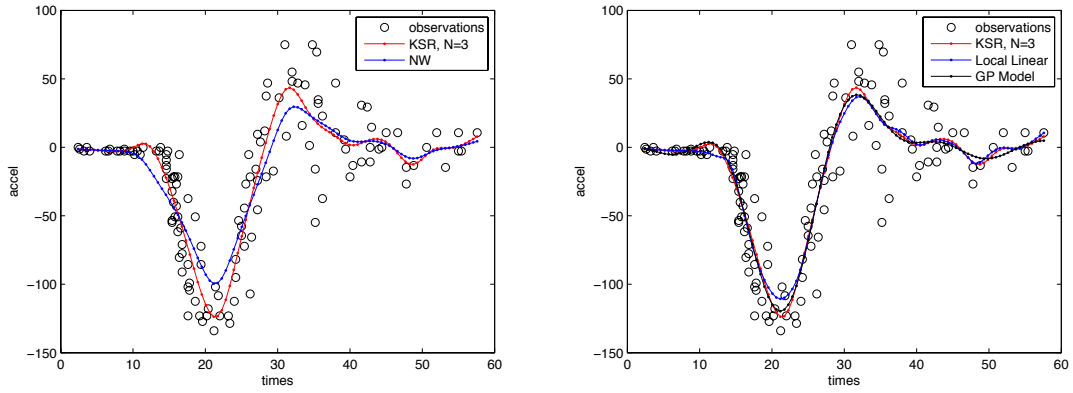


Figure 5.2.1: Compare KSR (N=3) prediction (red curve) with NW estimator (blue curve, left panel), local linear regression (blue curve, right panel), and Gaussian process model (black curve, right panel)

Table 5.2.1: Root mean square leave-one-out cross validation error.

Method	KSR (N=3)	NW (N=1)	Local Linear	GP Model
RMSCV	23.8553	26.4808	23.6926	23.3094

the other three methods, the NW estimator, the local linear regression, and the Gaussian process model (GP). The NW estimator has the same optimal bandwidth value as θ_1 for KSR. As shown in Figure 5.2.3, the NW estimator is clearly the worst among the four methods, since it is too smooth thus has missed some local variations. The KSR, local linear regression, and GP model have very similar fittings. But the KSR fits the peaks better than the other two methods, and both KSR and local linear regression has more local variations than the GP model. To show their performance numerically, we compared the leave-one-out cross-validation errors for the four methods in Table 5.2.3. The KSR method returns very similar RMSCV value (only slightly larger) as the local linear regression and the GP model. It clearly outperforms the NW estimator, which indicates the extra two regressions do improve the prediction accuracy.

Example 5.2.2. *Consider the test function used by Currin et al. (1991):*

$$y(x_1, x_2) = \{1 - \exp(-.5/x_2)\}\frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20} + \epsilon, \quad \boldsymbol{x} \in [0,1]^2.$$

We generate data using a 6×6*-grid design in* $[0,1]^2$*. The random noise follows normal*

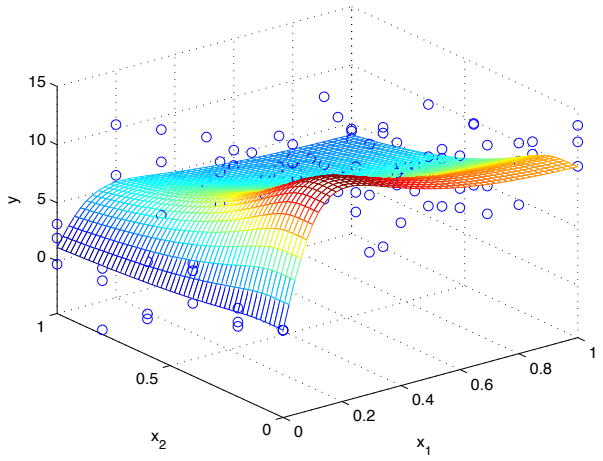


Figure 5.2.2: The true test function surface and observations.

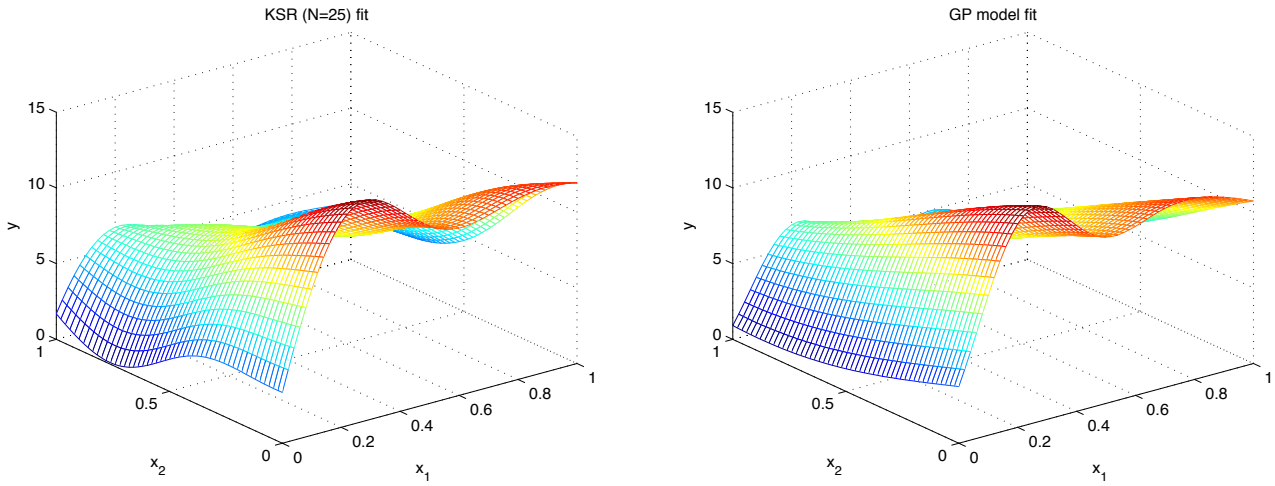


Figure 5.2.3: The KSR (N=25) fitted surface (left) and the GP model fitted surface (right).

distribution, $\epsilon \sim N(0, \sigma^2)$*, with non-constant variance* $\sigma = .25(\max(y) - \min(y)) \sin(x_1 + 2 * x_2)$*, which is also a function of* (x_1, x_2)*. Three replications are simulated for each design point. The function surface and the observations are shown in the left panel of Figure 5.2.3.*

Using Algorithm 5.2.1, the optimal value of N is 25. As shown in Figure 5.2.3, both the KSR and GP model capture the major nonlinearity of the true surface, but the GP model returns a more smooth surface while the KSR fitting has more local variation. We also compute the root mean square prediction error using the test data set of a 51×51-grid design in $[0, 1]^2$ and compare the KSR with the NW estimator, local linear regression, and

Table 5.2.2: Root mean square prediction error.

Method	KSR (N=25)	NW (N=1)	Local Linear	GP Model
RMSPE	0.7977	1.3474	0.8521	0.8845

GP model as in Example 2.1. The values of RMSPE are shown in Table 5.2.3. As the same in Example 2.1, the NW estimator still has the worst prediction error. The KSR performs the best among the four methods. The GP model does not perform as well as the KSR for this example, which is expected since the non-constant variance violates the stationary Gaussian process assumption.

5.3 Kernel Sum Interpolation

In the previous section, we have discussed the effect of N on the kernel sum regression. If we keep increasing N, the prediction accuracy does not keep increasing. The reason is that keep increasing N will reduce the bias of the prediction, but will meanwhile increase the variance, and the total mean squared error (MSE) might be increased as a result. But when there is no random noise contained in the response, which is very common in the field of computer experiments, there is no variance involved, thus large N reduces the MSE by reducing the bias. The toy example illustrated in Figure 5.3.1 shows that the kernel sum regression indeed improves the curve fitting by increasing N. As N increases, not only the fitted curve is closer to the true function, the predictions at the observation points x_i $i = 1, \ldots, n$ are also closer to the true observations $y(x_i)$. It is intuitive to conjecture that as N goes to infinity, the kernel sum regression method will become an interpolation method, i.e., $\hat{y}_N(x) = y(x_i)$ as $N \to \infty$ for $i = 1, \ldots, n$.

5.3.1 As $N \to \infty$

In fact we can show that the conjecture that the kernel sum regression converges to an interpolator as N goes to infinity is true. To prove this, we will need the following assumption for the kernel function.

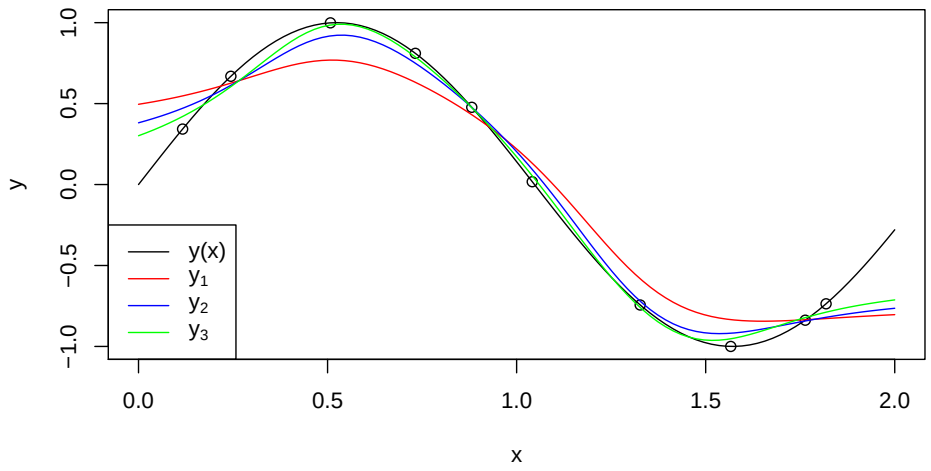


Figure 5.3.1: Comparison between the true function $y(x) = sin(3x)$ and KSR fitting $\hat{y}_1(x)$, $\hat{y}_2(x)$, and $\hat{y}_3(x)$.

Assumption 5.3.1. *The kernel function $K(\cdot,\cdot) : \Omega \times \Omega \to \mathbb{R}$ is a symmetric and strictly positive definite (s.p.d.) kernel on Ω, that is*

$$K(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{y}, \boldsymbol{x}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega$$

and

$$\sum_{i=1}^{n} \sum_{i=1}^{n} c_i c_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$$

for any $n > 0$ and any choice of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \Omega$, and any $c_1, \ldots, c_n \in \mathbb{R}$. The equality stands only when $c_i = 0$ for all i. The symmetric matrix $\boldsymbol{K}$ with entries $\boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive definite.

The following proposition states that the kernel sum regression becomes an interpolation method if $N \to \infty$, and we name it *kernel sum interpolation* (KSI). The proof is given in the Appendix.

Proposition 5.3.1. *The kernel sum regression $\hat{y}_\infty(\boldsymbol{x}) = \left(\sum_{i=1}^{\infty} \boldsymbol{u}_i(\boldsymbol{x})' \prod_{j=0}^{i-1}(\boldsymbol{I}_n - \boldsymbol{U}_j)\right)\boldsymbol{y}$ interpolates all the observations $(\boldsymbol{x}_i, y_i)$ for $i = 1, \ldots, n$.*

This proposition certainly applies to the kernel sum interpolation when all the kernel functions used are the same

$$\hat{y}(\boldsymbol{x})_\infty = \left(\boldsymbol{u}(\boldsymbol{x})' \sum_{i=0}^{\infty} (\boldsymbol{I}_n - \boldsymbol{U})^i \right) \boldsymbol{y}. \tag{5.3.1}$$

But this case can be more directly shown by

$$\rho(\boldsymbol{I}_n - \boldsymbol{U}) = 1 - \min_{1 \le i \le n} \lambda_i(\boldsymbol{U}) < 1,$$

which is the necessary and sufficient condition for $\sum_{i=0}^{\infty} (\boldsymbol{I}_n - \boldsymbol{U})^i = (\boldsymbol{I}_n - (\boldsymbol{I}_n - \boldsymbol{U}))^{-1} = \boldsymbol{U}^{-1}$, thus this kernel sum interpolation is simply

$$\hat{y}_\infty(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})' \boldsymbol{U}^{-1} \boldsymbol{y}. \tag{5.3.2}$$

A very similar numerical method called, "Iterated Approximate Moving Least Squares Approximation (IAMLSA)", has been studied by Fasshauer and Zhang (2006). Although the prediction forms of this method is very similar to that of kernel sum interpolation, they are developed based on very different perspectives. The former one is developed radial basis function interpolation, while the latter one is from kernel sum regression.

5.3.2 Connections with RBF, Kriging, and RIDW

The idea of kernel sum regression and interpolation is closely related to kernel regression. In fact, it also has some interesting connections with other interpolation methods, including radial basis function (RBF), kriging, and regression-based inverse distance weighting (RIDW) (Joseph and Kang, 2009).

Radial basis function interpolation takes the form

$$\hat{\boldsymbol{y}}_{RBF}(\boldsymbol{x}) = \sum_{i=1}^{n} c_k \varphi(\|\boldsymbol{x} - \boldsymbol{x}_k\|_2), \quad \boldsymbol{x} \in \Omega, \tag{5.3.3}$$

where $\varphi(\cdot)$ is a radial basis function and the coefficients c_k can be found by enforcing the interpolation conditions, and thus solving the linear system $\boldsymbol{\Psi c} = \boldsymbol{y}$. The matrix $\boldsymbol{\Psi}$ has

entries $\varphi(\|\boldsymbol{x}_j - \boldsymbol{x}_k\|_2)$, $j,k = 1,\ldots,n$. Let $\boldsymbol{r}(\boldsymbol{x}) = (\varphi(\|\boldsymbol{x} - \boldsymbol{x}_1\|_2),\ldots,\varphi(\|\boldsymbol{x} - \boldsymbol{x}_n\|_2))'$, thus the RBF interpolation can be simplified as

$$\hat{\boldsymbol{y}}_{RBF}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{\Psi}^{-1}\boldsymbol{y}. \tag{5.3.4}$$

The invertibility of $\boldsymbol{\Psi}$ can certainly be guaranteed if the radial basis function $\boldsymbol{\varphi}(\cdot)$ is a strictly positive definite radial basis function.

Kriging interpolation method takes essentially the same prediction form as the RBF method, but different from RBF, it is based on a stochastic assumption of the response y. Kriging assumes $y(\boldsymbol{x})$ follows the Gaussian process with the mean function $\mu(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})'\boldsymbol{\beta}$ and the covariance function $\sigma^2\phi(\boldsymbol{x}_1,\boldsymbol{x}_2)$ with the stationary variance σ^2, i.e., $y(\boldsymbol{x}) \sim GP(\mu(\boldsymbol{x}),\sigma^2\phi(\cdot))$. The prediction form of kriging is

$$\hat{y}_{Krig}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})'\hat{\boldsymbol{\beta}} + \boldsymbol{r}(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}), \quad \text{where } \hat{\boldsymbol{\beta}} = (\boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{R}^{-1}\boldsymbol{F}. \tag{5.3.5}$$

Here $\boldsymbol{f}(\boldsymbol{x})$ is a vector of functions of $\boldsymbol{x}$ and the rows of the matrix $\boldsymbol{F}$ are $\boldsymbol{f}(\boldsymbol{x}_i)$ $i = 1,\ldots,n$. The matrix $\boldsymbol{R}$ is the $n \times n$ correlation matrix with $R_{i,j} = \phi(\boldsymbol{x}_i,\boldsymbol{x}_j)$, and the vector are the correlations $r_i(\boldsymbol{x}) = \phi(\boldsymbol{x},\boldsymbol{x}_i)$. In the simplest case when the Gaussian process has zero mean, the prediction for is $\hat{y}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{y}$, which is the same as the RBF interpolation in (5.3.4). Therefore, disregarding the stochastic assumption, kriging and RBF are essentially the same interpolation method, and we consider kriging and RBF as the same type of interpolation methods.

The two type of interpolation methods, RBF (kriging) and KSI, have some interesting connections. Firstly, comparing the RBF interpolation in (5.3.4) and the KSI in (5.3.2), we can find that they have very similar prediction form. They are both linear predictor in terms of the observations $\boldsymbol{y}$, and they all involve inversion of an $n \times n$ matrix. Secondly, most of the strictly positive radial basis functions can be applied as kernel functions and correlation functions, such as Gaussian function, Matérn function, and generalized inverse multiquadrics, except that the radial basis function is isotropic. In Table 5.3.1 we compare these three radial basis functions and their corresponding kernel and correlation functions.

Table 5.3.1: Some radial basis functions and their corresponding kernel functions.

Function	RBF	Kernel/Correlation
Gaussian	$\exp(-\theta r^2)$	$\exp(-\sum_{i=1}^{d} \theta_i (x_{1,i} - x_{2,i})^2)$
Matérn	$\frac{B_{\beta-\frac{d}{2}}(r) r^{\beta-\frac{d}{2}}}{2^{\beta-1}\Gamma(\beta)}$	$\frac{B_{\beta-\frac{d}{2}}(\sqrt{\sum_{i=1}^{d} \theta_i (x_{1,i}-x_{2,i})}\{\sum_{i=1}^{d} \theta_i (x_{1,i}-x_{2,i})\}^{\frac{\beta}{2}-\frac{d}{4}}}{2^{\beta-1}\Gamma(\beta)}$
Generalized Inverse Multiquadrics	$(1 + r^2)^{-\beta}$	$(1 + \sum_{i=1}^{d} \theta_i (x_{1,i} - x_{2,i})^2)^{-\beta}$

$2\beta > d$, B_v is the modified Bessel function of the second kind of order v, and $r = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$.

Although RBF, kriging, and KSI have very similar prediction forms, they do have some differences. Suppose we use the same function for RBF and KSI and the bandwidth parameters, then the interpolation matrix $\boldsymbol{\Psi}$ is exactly the same as the matrix $\boldsymbol{K}$. In fact, $\boldsymbol{U} = \boldsymbol{S}^{-1}\boldsymbol{\Psi}$ ($\boldsymbol{S}$ is defined in the proof of Proposition 5.3.1) and $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x}) / \sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i)$. Therefore, the weight matrix $\boldsymbol{U}$ in KSI is a "normalized version" of $\boldsymbol{\Psi}$ because each row in $\boldsymbol{U}$ is the same row in $\boldsymbol{\Psi}$ divided by the sum of this row, and the vector $\boldsymbol{u}(\boldsymbol{x})$ is also a "normalized version" of $\boldsymbol{r}(\boldsymbol{x})$. The KSI becomes $\hat{y}_\infty = \boldsymbol{r}(\boldsymbol{x})\boldsymbol{\Psi}^{-1}(\boldsymbol{S} / \sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x}))\boldsymbol{y}$. Compared with RBF, KSI has an extra weight matrix ($\boldsymbol{S} / \sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x})$) on the responses $\boldsymbol{y}$. In Proposition 5.3.2, we show that the normalized version actually has advantage that the prediction is more robust to the choice of bandwidth.

Proposition 5.3.2. *Consider an exponential radial basis (correlation) function of the form* $\varphi(r) = \exp(-\theta r^q)$ *with* $\theta > 0$*, and kernel function* $K(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\|\boldsymbol{x} - \boldsymbol{y}\|_2)$*. Then, for the kernel sum interpolation, RBF, and ordinary kriging, we have*

$$\lim_{\theta\to\infty} \hat{y}_\infty(\boldsymbol{x}) = y_a, \quad \textit{where } a = arg \min_{a\in\{1,\ldots,n\}} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2,$$

$$\lim_{\theta\to\infty} \hat{y}_{RBF}(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \notin \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}, \\ y(\boldsymbol{x}), & \boldsymbol{x} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}, \end{cases}$$

$$\lim_{\theta\to\infty} \hat{y}_{OK}(\boldsymbol{x}) = \begin{cases} \hat{\mu}, & \boldsymbol{x} \notin \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}, \\ y(\boldsymbol{x}), & \boldsymbol{x} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}. \end{cases}$$

Figure 5.3.2 shows the advantage of the "normalization" of the kernel sum interpolation method. We fit both the KSI and the ordinary kriging to the true function $y(x) = \sin(2x)$ with 7 equally spaced sample points. Clearly, the KSI is uniformly better than the ordinary

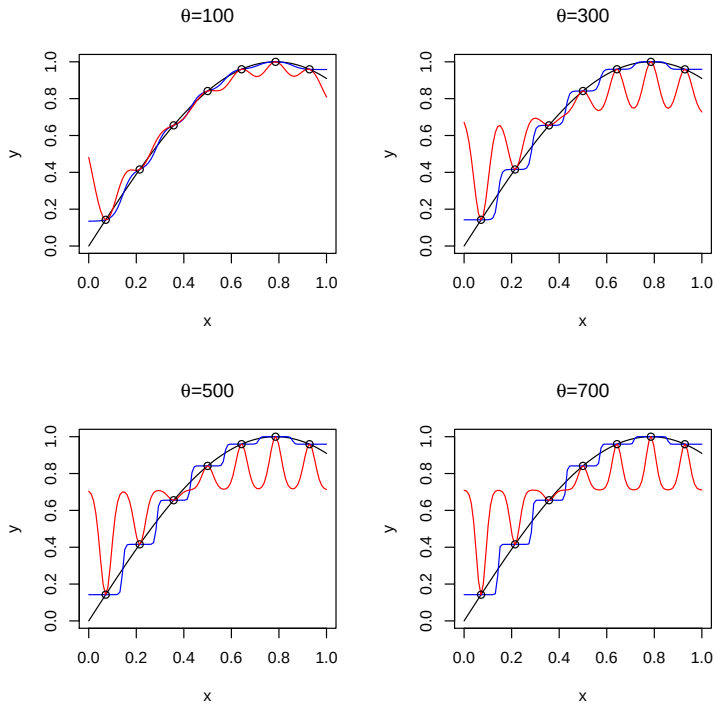


Figure 5.3.2: The true test function $y(x) = \sin(2x)$ (black), the ordinary kriging prediction (red), and the KSI prediction with $\theta = 100, 300, 500, 700$.

kriging. As θ becomes larger, the ordinary kriging prediction converges to the overall mean μ if the prediction points are not in the 7 sample points. The KSI predictions, on the other hand, converges to the sample observation $y(x_i)$ whose x_i is closest to the prediction points. Therefore, KSI is more robust to the bandwidth θ than the ordinary kriging predictor.

The regression-based inverse distance weighting method is a very different interpolation method. Its prediction form (5.3.6) does not need the matrix inversion to achieve interpolation.

$$\hat{y}_{RIDW}(\boldsymbol{x}) = \mu(\boldsymbol{x}, \boldsymbol{\theta}) + \frac{\sum_{i=1}^{n} w_i(\boldsymbol{x}_i, \boldsymbol{x}) e_i}{\sum_{i=1}^{n} w(\boldsymbol{x}_i, \boldsymbol{x})}, \tag{5.3.6}$$

$$\text{where } e_i = y_i - \mu(\boldsymbol{x}_i, \boldsymbol{\theta}) \quad \text{and } w(\boldsymbol{x}_i, \boldsymbol{x}) = \frac{\exp(-\sum_{i=1}^{d} \theta_i (x_{1,i} - x_i)^2)}{\sum_{i=1}^{d} \theta_i (x_{1,i} - x_i)^2}.$$

Instead, the interpolation is achieved through the weight function $w(\boldsymbol{x}_i, \boldsymbol{x})$. The RIDW interpolation is very general because the regression part $\mu(\boldsymbol{x}, \boldsymbol{\theta})$ can be linear, nonlinear, or nonparametric model. For instance, using NW estimator as the regression model, the

RIDW (5.3.7) can be called kernel-regression based inverse distance weighting.

$$\hat{y}_{RIDW}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} K_\theta(\boldsymbol{x}_i, \boldsymbol{x}) y_i}{\sum_{i=1}^{n} K_\theta(\boldsymbol{x}_i, \boldsymbol{x})} + \frac{\sum_{i=1}^{n} w_i(\boldsymbol{x}_i, \boldsymbol{x}) e_i}{\sum_{i=1}^{n} w_i(\boldsymbol{x}_i, \boldsymbol{x})}. \tag{5.3.7}$$

It is a special of kernel sum interpolation with $N = 2$, $K_1(\cdot, \cdot) = K_\theta(\cdot, \cdot)$, and $K_2(\cdot, \cdot) = w(\cdot, \cdot)$. Because of the special weight function $w(\cdot, \cdot)$, N does not need to be infinity to achieve interpolation.

5.4 Conclusions

In this chapter, we propose two modeling approach. The first one is kernel sum regression, which uses an iterative implementation of the simple classic kernel regression. An algorithm is constructed to choose the optimal number of regressions N and the bandwidth parameters based on the generalized cross-validation. The performance of the kernel sum regression is shown to be superior than the simple kernel regression through two examples, thus the extra regressions do improve the prediction. In the second part, we show that as the number of iterations increases to infinity, the kernel sum regression converges to an interpolator, which we name as kernel sum interpolation. It has many interesting connections with the other interpolation methods, such as radial basis function, kriging, as well as the regression-based inverse distance weighting method. Compared with these interpolators, kernel sum interpolation is shown to be more robust to the bandwidth parameter.

5.5 Appendix: Proofs

Proof of Proposition 5.3.1:

It is equivalent to show $\|\boldsymbol{y} - \hat{\boldsymbol{y}}_\infty\|_2 = 0$.

$$\begin{aligned} \boldsymbol{y} - \hat{\boldsymbol{y}}_\infty &= \left(\boldsymbol{I}_n - \sum_{i=1}^{\infty} \boldsymbol{U}_i \prod_{j=0}^{i-1} (\boldsymbol{I}_n - \boldsymbol{U}_j) \right) \boldsymbol{y} \\ &= \left(\prod_{i=1}^{\infty} (\boldsymbol{I}_n - \boldsymbol{U}_i) \right) \boldsymbol{y} \end{aligned}$$

For any matrix norm we have

$$\|\boldsymbol{y} - \hat{\boldsymbol{y}}_\infty\| \leq \prod_{i=1}^{\infty} \|\boldsymbol{I}_n - \boldsymbol{U}_i\| \cdot \|\boldsymbol{y}\|$$

Let $\rho(\cdot)$ be the spectral radius of a matrix, i.e., $\rho(\boldsymbol{X}) = max_{1\leq i\leq n}|\lambda_i(\boldsymbol{X})|$, where $\lambda_i(\boldsymbol{X})$ are the eigenvalues of matrix $\boldsymbol{X}$. It is known that for any matrix norm $\|\cdot\|$, we have $\rho(\boldsymbol{X}) \leq \|\boldsymbol{X}\|$. Here $\|\boldsymbol{U}_i\|_\infty = 1$, so $\rho(\boldsymbol{U}_i) \leq \|\boldsymbol{U}_i\|_\infty = 1$. Because $\boldsymbol{U}_i = \boldsymbol{S}_i^{-1}\boldsymbol{K}_i$, where $\boldsymbol{K}_i$ is a sysmetric positive definite matrix due to Assumption 5.3.1 and $\boldsymbol{S}_i$ is a diagonal matrix with all positive diagonal entries. We have

$$0 < \rho(\boldsymbol{U}_i) \leq 1 \quad \Rightarrow 0 < \min_{1\leq j\leq n} \lambda_j(\boldsymbol{U}_i) \leq 1 \Rightarrow 0 \leq 1 - \min_{1\leq j\leq n} \lambda(U_i) < 1.$$

Since $\|\boldsymbol{I}_n - \boldsymbol{U}_i\|_2$ is the maximum absolute eigenvalue of $(\boldsymbol{I}_n - \boldsymbol{U}_i)'(\boldsymbol{I}_n - \boldsymbol{U}_i)$, which are equal to $(1 - \min_{1\leq j\leq n} \lambda_j(\boldsymbol{U}_i))^2 < 1$, $\|\boldsymbol{I}_n - \boldsymbol{U}_i\|_2 < 1$ for all $i = 1, 2, \ldots$. Therefore, we have $\prod_{i=1}^{N} \|\boldsymbol{I}_n - \boldsymbol{U}_i\| \to 0$ as $N \to \infty$ and

$$\|\boldsymbol{y} - \hat{\boldsymbol{y}}_\infty\|_2 \leq \prod_{i=1}^{\infty} \|\boldsymbol{I}_n - \boldsymbol{U}_i\|_2 \cdot \|\boldsymbol{y}\|_2 = 0.$$

Proof of Proposition 5.3.2:

The result for ordinary kriging $\hat{y}_{OK}(\boldsymbol{x})$ has been proved by Joseph (2006). Since $\hat{y}_{RBF}(\boldsymbol{x})$ is just $\hat{y}_{OK}(\boldsymbol{x})$ with $\mu = 0$, thus the result for RBF is also true. We only need to show the result for the KSI. As explained in Section 5.3.2, under the assumption in Proposition 5.3.2, the KSI predictor can be written as

$$\hat{y}_\infty(\boldsymbol{x}) = \frac{\boldsymbol{r}(\boldsymbol{x})}{\sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x})} \boldsymbol{\Psi}^{-1}\boldsymbol{S}\boldsymbol{y},$$

where the ith element of the vector $\frac{\boldsymbol{r}(\boldsymbol{x})}{\sum_{i=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x})}$ is

$$\frac{\exp\{-\theta\|\boldsymbol{x} - \boldsymbol{x}_i\|^q\}}{\sum_{j=1}^{n} \exp\{-\theta\|\boldsymbol{x} - \boldsymbol{x}_j\|^q\}}. \tag{5.5.1}$$

If we denote i^* as the index such that $\|\boldsymbol{x} - \boldsymbol{x}_{i^*}\| = \min_j \|\boldsymbol{x} - \boldsymbol{x}_j\|$ and

$$\delta_{i^*}(i) = \exp\{-\theta(\|\boldsymbol{x} - \boldsymbol{x}_i\|^q - \|\boldsymbol{x} - \boldsymbol{x}_{i^*}\|^q)\},$$

then (5.5.1) can be equivalently written as

$$\frac{\exp\{-\theta\|\boldsymbol{x}-\boldsymbol{x}_{i^*}\|^q\}\delta_{i^*}(i)}{\exp\{-\theta\|\boldsymbol{x}-\boldsymbol{x}_{i^*}\|^q\}\sum_{j=1}^n \delta_{i^*}(j)} = \frac{\delta_{i^*}(i)}{\sum_{j=1}^n \delta_{i^*}(j)}.$$

As $\theta \to \infty$,

$$\frac{\delta_{i^*}(i)}{\sum_{j=1}^n \delta_{i^*}(j)} \to 1 \text{ if } i = i^*, \text{ otherwise } \frac{\delta_{i^*}(i)}{\sum_{j=1}^n \delta_{i^*}(j)} \to 0.$$

Therefore, $\boldsymbol{r}(\boldsymbol{x})\sum_{i=1}^n K(\boldsymbol{x}_i,\boldsymbol{x}) \to \boldsymbol{v}_{i^*}$, where $\boldsymbol{v}_{i^*}$ is the vector whose i^*th element equal to 1 and others equal to 0. Moreover, as $\theta \to \infty$, it is easy to see that $\boldsymbol{S}^{-1}\boldsymbol{\Psi} = \boldsymbol{I}_n$. Thus $\hat{y}_\infty(\boldsymbol{x}) \to y_{i^*}$ as $\theta \to \infty$.

REFERENCES

Ababou, R., Bagtzoglou, A. C., and Wood, E. F. (1994). On the Condition Number of Covariance Matrices in Kriging, Estimation, and Simulation of Random Fields. *Mathematical Geology*, **26**, 99-133.

Ai, M.Y., Zhang, R.C. (2004). Theory of optimal blocking of nonregular factorial designs. *The Canadian Journal of Statistics*, **32**(1), 57-72.

An, J., and Owen, A. (2001). Quasi-Regression. *Journal of Complexity*, **17**, 588-607.

Bates, R. A., Giglio, B., and Wynn, H. P. (2003). A Global Selection Procedure for Polynomial Interpolators. *Technometrics*, **45**, 246-255.

Bingham, D., and Sitter, R. R. (2003). Fractional Factorial Split-Plot Designs for Robust Parameter Experiments. *Technometrics*, **45**, 80-89.

Bisgaard, S. (1994). A note on the definition of resolution for blocked 2^{k-p} designs. *Technometrics*, **36**, 308–311.

Borges, C., Bruns, E. R., Almeida, A. A., and Scarminio, I. S. (2007). Mixture-mixture design for fingerprint optimization of chromatographic mobile phases and extraction solutions for Camellia sinensis. *Analytica Chimica Acta*, **595**, 28-37.

Borror, C. M., Montgomery, D. C., and Myers, R. H. (2002). Evaluation of Statistical Designs for Experiments Involving Noise Variables. *Journal of Quality Technology*, **34**, 54-70.

Chen, J., Sun, D.X. and Wu, C.F.J. (1993). A catalogue of two-level and three-level fractional factorial designs with small runs. *International Statistical Review*, **61**, 131-145.

Chen, H. and Cheng, C.S. (1999). Theory of optimal blocking of 2^{n-m} designs. *Annals of Statistics*, **27**, 1948-1973.

Cheng, S.W. and Wu, C.F.J. (2002). Choice of optimal blocking schemes in two-level and three-level designs. *Technometrics*, **44**, 269-277.

Cornell, J. A., and Good, I. J. (1970). The Mixture Problem for Categorized Components. *Journal of the American Statistical Association*, **65**, 339-355.

Cornell, J. A. (1971). Process Variables in the Mixture Problem for Categorized Components. *Journal of the American Statistical Association*, **66**, 42-48.

Cornell, J. A., and Ramsey, P. J. (1998). A Generalized Mixture Model for Categorized-Components Problems With an Application to a Photoresist-Coating Experiment. *Technometrics*, **40**, 48-61.

Cornell, J. A. (2002), *Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data* (3rd ed.), New York: Wiley.

Cornell, J. A., and Gorman, J. W. (2003). Two New Mixture Models: Living Collinearity but Removing Its Influence. *Journal of Quality Technology*, **35**, 78-88.

Currin, C., Mitchell, T. J., Morris, M. D., and Ylvisaker, D. (1991). Bayesian Prediction of Deterministic Functions with Applications to the Design and Analysis of Computer Experiments. *Journal of the American Statistical Association*, **86**, 953-963.

Davis, G. J., and Morris, M. D. (1997). Six Factors Which Affect the Condition Number of Matrices Associated with Kriging. *Mathematical Geology*,**29**, 669-683.

Del Castillo, E., Alvarez, M. J., Ilzarbe, L., and Viles, E. (2007). A New Design Criterion for Robust Parameter Experiments. *Journal of Quality Technology*, **39**, 279-295.

Didier, C., Etcheverrigaray, M., Kratje, R., and Goicoechea, H. C. (2007). Crossed mixture design and multiple response analysis for developing complex culture media used in recombinant protein production. *Chemometrics and Intelligent Laboratory Systems*, **86**, 1-9.

Dingstad, G., Egelandsdal, B., and NæS, T. (2003). Modeling methods for corssed mixture experiments—a case study from sausage production. *Chemometrics and intelligent laboratory systems*, **66**, 175-190.

Efron, B., Johnstone, I., Hastie, T., and Tibshirani, R. (2004). Least Angle Regression (with discussion). *The Annals of Statistics*, **32**, 407-499.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and its Applications*, Chapman and Hall: London.

Fasshauer, G. E. and Zhang, J. G. (2007). Iterated Approximate Moving Least Squares Approximation. *Advances in Meshfree Techniques* (Leitao, M. A., Alves, C., and Duarte, C. A. (eds.)) Springer, 221-240.

Franke, R. (1979). A critical comparison of some methods for interpolation of scattered data. *Technical Report NPS-53-79-003*, Department of Mathematics, Naval Postgraduate School, Monterey, California.

Franke, R., and Nielson, G. (1980). Smooth interpolation of large sets of scattered data. *Journal for Numerical Methods in Engineering*, **15**, 1691-1704.

Freeny, A. E. and Nair, V. N. (1992). Robust Parameter Design with Uncontrollable Noise Variables. *Statistica Sinica*, **2**, 313-334.

Gasser, T. and Müller, H-G. (1984). Estimating regression of functions and their derivatives by the kernel method. *Scandinavian Journal of Statistics*, 11, 171-185.

Ginsburg, H., and Ben-Gal, I. (2006). Designing experiments for robust-optimization problems: the Vs-optimality criterion. *IIE Transactions on Quality and Reliability Engineering*, **38**, 445-461.

Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21, 215-223.

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian Treed Gaussian Process Models With an Application to Computer Modeling. *Journal of the American Statistical Association*, **103**, 1119-1130.

Hung, Y., Joseph, V. R., and Melkote, S. N. (2009). Design and Analysis of Computer Experiments with Branching and Nested Factors. *Technometrics*, **51**, 354-365.

Joseph, V. R. (2003). Robust Parameter Design with Feed-Forward Control. *Technometrics*, **45**, 284-292.

Joseph, V. R. (2006). A Bayesian Approach to the Design and Analysis of Fractionated Experiments. *Technometrics*, **48**, 219-229.

Joseph, V. R. (2007). Taguchi's Approach to Robust Parameter Design: A New Perspective. *IIE Transactions on Quality and Reliability Engineering*, **39**,805-810.

Joseph, V. R. (2008). Tolerance Design. *Encyclopedia of Statistics in Quality and Reliability* (Eds. Ruggeri, F., Kennet, R. S., and Faltin, F.), New York: Wiley, 2014-2019.

Joseph, V. R., and Delaney, J. D. (2007). Functionally Induced Priors for the Analysis of Experiments. *Technometrics*, **49**, 1-11.

Joseph, V. R., Hung, Y., and Sudjianto, A. (2008). Blind Kriging: A New Method for Developing Metamodels. *ASME Journal of Mechanical Design*, **130**, 031102-1-8.

Kang, L. and Joseph, V. R. (2009). Bayesian Optimal Single Arrays for Robust Parameter Design. *Technometrics*, **51**, 250-261.

Lambrakis, D. P. (1968). Experiments with Mixtures: A Generalization of the Simplex-Lattice Design. *Journal of the Royal Statistical Society Series B*, **30**, 123-136.

Lambrakis, D. P. (1969). Experiments with Mixtures: Estimated Regression Function of the Multiple-Lattice Design. *Journal of the Royal Statistical Society Series B*, **31**, 276-284.

Lazzaro, D. and Montefusco, L. B. (2002). Radial basis functions for the multivariate interpolation of large scattered data sets. *Journal of Computational and Applied Mathematics*, **140**, 521-536.

Li, R. and Sudjianto, A. (2005). Analysis of Computer Experiments Using Penalized Likelihood in Gaussian Kriging Models. *Technometrics*, **47**, 111-120.

Li, W. W., and Wu, C. F. J. (1997). Columnwise-Pairwise Algorithms With Applications to the Construction of Supersaturated Designs. *Technometrics*, **39**, 171-179.

Li, X., Sudarsanam, N., and Frey, D. D. (2006). Regularities in Data from Factorial Experiments. *Complexity*, **11**, 32-45.

Liszka, T. (1984). An interpolation method for an irregular net of nodes. *International Journal for Numerical Methods in Engineering*, **20**, 1599-1612.

Łukaszyk, S. (2004). A new concept of probability metric and its applications in approximation of scattered data sets. *Computational Mechanics*, **33**, 299-304.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, **21**, 239-245.

Meyer, R. K., and Nachtsheim, C. J. (1995). The Coordinate-Exchange Algorithm for Constructing Exact Optimal Experimental Designs. *Technometrics*, **37**, 60-69.

Miller, A., Sitter, R., Wu, C. F. J., and Long, D. (1993). Are Large Taguchi-style Experiments Necessary? A Reanalysis of Gear and Pinion Data. *Quality Engineering*, **6**, 21-37.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory Designs for Computer Experiments. *Journal of Statistical Planning and Inference*, **43**, 381-402.

Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability and Its Applications*, 9, 141-142.

Nguyen, N., and Miller, A. J. (992). A Review of Some Exchange Algorithm for Constructing Discrete D-optimal Designs. *Computational Statistics & Data Analysis*, **14**, 489-498.

Piepel, G. F. (1988). Programs for Generating Extreme Vertices and Centroids of Linearity Constrained Experimental Regions. *Journal of Quality Technology*, **20**, 125-139.

Piepel, G. F. (1999). Modeling Methods for Mixture-of-Mixtures Experiments Applied to a Tablet Formulation Problem. *Pharmaceutical Development and Technology*, **4**, 593-606.

Prescott, P., Dean, A. M., Draper, N. R., and Lewis, S. M. (2002). Mixture Experiments: ILL-Condioning and Quadratic Model Specification. *Technometrics*, **44**, 260-268.

Renka, R. J. (1988). Multivariate Interpolation of Large Sets of Scattered Data. *ACM Transactions on Mathematical Software*, **14**, 139-148.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, **4**, 409-435.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 ACM National Conference*, 517-524.

Shoemaker, A. C., Tsui, K. L. and Wu, C. F. J. (1991). Economical experimentation methods for robust design. *Technometrics*, **33**, 415-427.

Sitter, R.R., Chen, J., Feder, M. (1997). Fractional resolution and minimum aberration in blocked 2^{n-k} design. *Technometrics*, **39**, 382-390.

Snee R. D., and D. W. Marquardt (1974). Extreme Vertices Designs for Linear Mixture Models. *Technometrics*, **16**, 399-408.

Snee R. D. (1975). Experimental Designs for Quadratic Models in Constrained Mixture Spaces. *Technometrics*, **17**, 149-159.

Suen, C.Y., Chen, H., Wu, C.F.J. (1997). Some identities on q^{n-m} designs with application to minimum aberrations. *Annals of Statistics*, **25**, 1176-1188.

Sun, D.X., Wu, C.F.J., Chen, Y. (1997). Optimal blocking schemes for 2^n and 2^{n-p} designs. *Technometrics*, **39**, 298-307.

Taguchi, G. (1987), *System of Experimental Design*, Vol 1 & 2, White Plains, New York: Unipub/Kraus International.

Tang, B., Wu, C.F.J. (1996). Characterization of minimum aberration 2^{n-k} designs in terms of their complementary designs. *Annals of Statistics*, **25**, 1176-1188.

Watson, G. S. (1964). Smooth regression analysis. *Sankhya Series A*, 26, 359-372.

Welch, W. J., Yu, T. K., Kang, S. M. and Sacks, J. (1990).Computer experiments for quality control by parameter design. *Journal of Quality Technology*, **22**, 15-22.

Wu, C. F. J., and Hamada, M. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*, New York: Wiley.

Wu, C. F. J., and Zhu, Y. (2003). Optimal Selection of Single Array for Parameter Design Experiments. *Statistica Sinica*, **13**, 1179-1199.

Xu, H. (2006), Blocked Regular Fractional Factorial Designs With Minimum Aberration. *Annals of Statistics*, **34**, 2534-2553.

Xu, H., Lau, S. (2006). Minimum Aberration Blocking Schemes for Two- and Three-Level Fractional Factorial Designs. *Journal of Statistical Planning and Inference*, **136**, 4088-4118.

Yamamoto, J. K. (2000). An Alternative Measure of the Reliability of Ordinary Kriging Estimates. *Mathematical Geology*, **32**, 489-509.

Yuan, M., Joseph, V. R., and Lin, Y. (2007). An Efficient Variable Selection Approach for Analyzing Designed Experiments. *Technometrics* **49**, 430-439.

Zhang, R., Park, D.K. (2000). Optimal blocking of two-level fractional factorial designs. *Journal of Statistical Planning and Inference*, **91**, 107-121.

Zhu, Y., Zeng, P. (2005). On the coset pattern matrix and minimum M-aberration of 2^{n-p} designs. *Statistics Sinica*, **15**, 717-730.

Zhu, Y., Zeng, P., and Jennings, K. (2007). Optimal Compound Orthogonal Arrays and Single Arrays for Robust Parameter Design Experiments. *Technometrics*, **49**, 440-453.